

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**

Requested Patent: JP3046043A

Title:

METHOD FOR ALLOCATING REAL PAGES TO VIRTUAL PAGES HAVING
DIFFERENT PAGE SIZES THEREFROM. ;

Abstracted Patent: EP0408070, A3, B1 ;

Publication Date: 1991-01-16 ;

Inventor(s):

KAGIMASA TOYOHICO (JP); MORI TOSHIAKI (JP); MATSUDA YOSHIKI (JP);
TAKAHASI KIKUO (JP) ;

Applicant(s): HITACHI LTD (JP) ;

Application Number: EP19900113477 19900713 ;

Priority Number(s): JP19890180323 19890714 ;

IPC Classification: G06F12/14 ;

Equivalents: DE69033064D, DE69033064T, JP2858795B2

ABSTRACT:

In a real storage allocation method, a real storage containing a plurality of real pages to which storage keys used for a storage protection purpose should be stored is allocated to a plurality of virtual pages of a virtual storage in accordance with the real pages. The real storage allocation method includes the steps of: (a) allocating one of plural split regions each having a first size which is obtained by subdividing one of plural real pages each having a second size equal to a value "n" ("n" being a positive integer) times higher than the first size, and which is provided with the real storage, to one virtual page having the first size employed to the virtual storage, and (b) repeating the step (a) in such a manner that one of plural split regions each having a first size which is obtained by subdividing other split regions within the one real page, or other real pages having the second sizes, is allocated to other virtual pages having the first pages. Furthermore, the real storage allocation method includes the steps of: (c) among a plurality of first sort of virtual pages each having a first size and a plurality of second sort of virtual pages each having a second size $1/n$ ("n" being a positive integer) times the first size, which have been previously prepared, when the real storage is allocated to the first sort of virtual page, allocating to the first sort of virtual page, a real page group constructed of one of the first sort of real pages, or "n" pieces of the second sort of continuous real pages which have not yet been allocated to the virtual page among a plurality of first sort of real storages each having the first virtual pages previously employed in the real storage and also a plurality of second sort of real pages each having the second size; and (b) when the real storage is allocated to the second sort of virtual page, allocating one of "n" pieces of split regions obtained by dividing one of the first sort of real page, or one of the second sort of real page.

⑨ 日本国特許庁(JP)

⑩ 特許出願公開

⑫ 公開特許公報(A)

平3-46043

⑬ Int.Cl.³

G 06 F 12/08

識別記号

W

庁内整理番号

7232-5B

⑭ 公開 平成3年(1991)2月27日

審査請求 未請求 請求項の数 21 (全30頁)

⑮ 発明の名称 実記憶割り当て方法

⑯ 特 願 平1-180323

⑰ 出 願 平1(1989)7月14日

⑱ 発 明 者 高 橋 喜 久 雄 東京都国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中央研究所内
⑱ 発 明 者 鍵 政 豊 彦 東京都国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中央研究所内
⑱ 発 明 者 松 田 芳 樹 東京都国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中央研究所内
⑱ 発 明 者 森 利 明 東京都国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中央研究所内
⑲ 出 願 人 株式会社日立製作所 東京都千代田区神田駿河台4丁目6番地
⑲ 代 理 人 弁理士 小川 勝男 外1名

明 細 書

1. 発明の名称

実記憶割り当て方法

2. 特許請求の範囲

1. それぞれに対応して記憶保護のための記憶キーが記憶されるべき複数の実ページを含む実記憶を、仮想記憶の複数の仮想ページに割り当てる方法において、(a) 第1のサイズの複数の仮想ページおよび該第1のサイズのn(nは正整数)倍の第2のサイズの複数の実ページをそれぞれ仮想記憶および実記憶に準備し、(b) 各実ページを分割して得られる該第1のサイズを有する複数の分割領域をそれぞれ異なる仮想ページに割り当てることを特徴とする実記憶割り当て方法。
2. (c) 該複数の実ページの一つに含まれる分割領域がすべて未割り当てのときに、その一つの分割領域をあるプログラムが使用する仮想ページに割り当てるとき、該仮想ページにその時与えられている記憶キーをもってその実ページ

をアクセスするのを許可する値を有する記憶キーを該一つの実ページに対する記憶キーとして記憶するステップをさらに有する第1項の実記憶割り当て方法。

3. 該ステップにおいて、少くとも一つの分割領域が仮想ページに割り当て済みで少くとも一つの他の分割領域が仮想ページに割り当てられていない部分未割り当ての実ページであって、実記憶を割り当てるべき仮想ページを使用するプログラムにその時与えられたアクセスキーをもってその実ページをへアクセスするのを許可する値を、該ページに対応して記憶された記憶キーが有する実ページを捜し、該一つの実ページ中の該他の分割領域を該実記憶を割り当てるべき仮想ページに割り当てる第2項の実記憶割り当て方法。
4. 該一つの割り当て済みの分割領域が該プログラムが使用する仮想ページのいずれかに割り当てられた分割領域である第3項の実記憶割り当て方法。

5. 該実ページを捜しえないとき、いずれの仮想ページに対しても未割り当ての分割領域のみからなる他の実ページを捜し、それらの分割領域の一つを該実記憶を割り当てるべき仮想ページに割り当てる第3項の実記憶割り当て方法。
6. いずれかの仮想ページに割り当て済みの分割領域を有する複数の実ページの各々について、それに含まれる分割領域のいずれかがプログラムにより参照されたか否かに依存してそれぞれの実ページがプログラムにより参照されたか否かを計測するステップをさらに有する第1項の実記憶割り当て方法。
7. それぞれの実ページに含まれる複数の分割領域を一緒にページアウトするステップをさらに有する第1項の実記憶割り当て方法。
8. 該第1のサイズと同じサイズの複数の実ページをさらに該実記憶にあらかじめ準備し、該第2のサイズを有する複数の実ページの一つにある一つの分割領域と該第1のサイズを有する複数の実ページ内、指定によりかわる一方を選択する第12項の実記憶割り当て方法。
9. 該第1のサイズを有する複数の実ページをさらに該実記憶にあらかじめ準備し、該ステップ(b)において、(b1) 該第1のサイズを有する複数の実ページの中に実記憶の割り当てを要求する仮想ページに対して割り当て可能な少なくとも一つの分割領域があるかをあらかじめ定めた割当て条件により判別し、(b2) 該一つの分割領域がある場合には、その分割領域を該仮想ページに割り当て、(b3) 該一つの分割領域がない場合には該第1のサイズを有する該複数の実ページの一つを該仮想ページに割り当てる第1項の実記憶割り当て方法。
10. それぞれに対応して記憶保護のための記憶キーが記憶されるべき複数の実ページを有する実記憶内の領域を仮想記憶の複数の仮想ページに割り当てる方法において、(a) 第1のサイズの複数の仮想ページおよび該第1のサイズを有する第12項の実記憶割り当て方法。
11. 該一つの仮想ページを使用するプログラムにその時与えられた仮想ページの記憶キーをもってそのn個の実ページの群の各々の実ページをアクセスするのを許可する値の記憶キーを該n個の実ページの各々に対応して記憶するステップをさらに有する第11項の実記憶割り当て方法。
12. 各実ページ群について、そこに含まれる実ページのいずれかがプログラムにより参照されたかに依存して、それぞれの実ページ群が参照されたか否かを計測するステップをさらに有する第11項の実記憶割り当て方法。
13. それぞれの実ページ群に含まれる実ページを一緒にページアウトするステップをさらに有する第12項の実記憶割り当て方法。
14. 該ステップ(a)において、該第2のサイズを有する複数の仮想ページをさらに該仮想記憶にあらかじめ準備し、該ステップ(b)において実記憶を割り当てるべき第1の仮想ページが該第1のサイズを有する場合、該複数の実ページ群の内、仮想ページへの割り当てに用いられていないn個のページのみからなる第1の群内の実ページを該第1の仮想ページに割り当て、実記憶を割り当てるべき第2の仮想ページが該第2のサイズを有する場合、該複数の実ページ群の第2の群に属し、n個の実ページの内、仮想ページの割り当てに用いられていない一つの実ページに割り当てる第10項の実記憶管理方法。
15. 該第1の仮想ページに実ページを割り当てた時に、そのときに与えられていたアクセスキーに実記憶へのアクセスを許可する値の記憶キーを、該第1の実ページ群に含まれるn個の実ページの各々に対応して記憶し、該第2の仮想

ページに実ページを割り当てた時に、与えられていたアクセスキーに対して実記憶へのアクセスを許可する値の記憶キーを、該第2の実ページ群内の n 個の実ページの内、該一つの実ページに対応して記憶するステップをさらに有する第14項の実記憶割り当て方法。

16. 該複数の実ページ群の内、該第1のサイズを有する仮想ページへ割り当てられた複数の第1種実ページ群の各群について、その実ページ群がプログラムが参照されたか否かを、その実ページ群内のいずれかの実ページが参照されたか否かに依存して計測し、該複数の実ページの内、該第2のサイズを有する仮想ページに割り当てられた少くとも一つの実ページを有する第2種の実ページ群にいずれかに属する実ページの各々について、それぞれの実ページがプログラムにより参照されたか否かを計測するステップをさらに有する第14項の実記憶管理方法。

17. 該第1種の実ページ群の各々に含まれる n 個の実ページを一緒にページアウトし、該第2

種の実ページ群の各々に含まれる n 個の実ページのそれぞれを他の実ページとは別にページアウトするステップをさらに有する第15項の実記憶管理方法。

18. (d) 該第2のサイズを有する複数の第3種の実ページを該実記憶にさらに設け、(e) 実記憶を割り当てるべき第2種の仮想ページに対して、該第3種の実ページの内、仮想ページに割り当てられていない一つを捜し、(f) 該一つの第3種の実ページが検出されたときには、該実記憶を割り当てるべき第2種の仮想ページに該検出された一つの第3種の実ページを割り当て、(g) 該一つの第3種の実ページが検出されなかったときに該ステップ(c)を実行する。

19. 該ステップ(c)は、(c1) 仮想ページに未だ割り当てられていない第2種の実ページが少くとも一つあるかを判別し、(c2) 該一つの第2種の実ページがあれば、それを実記憶割り当て対象の該仮想ページに割り当て、(c

3) 該一つの第2種の実ページがなければ、該連続する n 個の第1種の実ページを割り当てるステップからなる第16項の実記憶割り当て方法。

20. それぞれに対応して記憶保護のための記憶キーが記憶されるべき複数の実ページを有する記憶を仮想記憶の複数の仮想ページに割り当てる実記憶割り当て方法において、(a) 第1のサイズの複数の第1種の仮想ページと該第1のサイズの n 倍(n は正整数)を有する第2のサイズの複数の第2種の仮想ページとをあらかじめ仮想記憶に準備し、(b) 該第1のサイズを有する複数の第1種の実記憶と該第2のサイズを有する複数の第2種の実ページとを実記憶にあらかじめ準備し、(c) 実記憶を割り当てるべき第1種の仮想ページに対して、該第1種の実ページの一つ又は該第2種の実ページの一つを分割して得られる n 個の分割領域の一つを割り当て、(d) 実記憶を割り当てるべき第2種の仮想ページに対して、該第2種の実ページの

一つ又は連続する n 個の、仮想ページにまだ割り当てられていない第1種の実ページを割り当てる実記憶割り当て方法。

21. ステップ(b)において該第1のサイズを有する第3種の複数の実ページおよび該第2のサイズを有する第4種の複数の実ページをあらかじめ該実記憶に準備し、ステップ(c)において、(c1) 実記憶を割り当てるべき第1種の仮想ページに対して割り当て可能な少くとも一つの第3種の実ページがあるかをあらかじめ定めた第1の条件により判別し、(c2) 該一つの第3種の実ページがある場合、それら該第1種の仮想ページに割り当て、(c3) 該一つの第3種の実ページがない場合、該第1種の仮想ページに割り当て可能な少くとも一つの第1種の実ページがあるかをあらかじめ定めた第2の条件により判別し、(c4) 該一つの第1種の実ページがある場合、それを該第1種の仮想ページに割り当て、ステップ(d)において、(d1) 実記憶を割り当てるべき第2種の仮想

ページに対して割り当て可能な少くとも一つの第4種の実ページがあるかをあらかじめ定めた第3の条件により判別し、(d2)該一つの第4種の実ページがある場合は、それを該第2種の仮想ページに割り当て、(d3)該一つの第4種の実ページがない場合は、該第2種の仮想ページに割り当て可能な少くとも一つの第2種の実ページがあるかをあらかじめ定めた第4の条件により判別し、(d4)該一つの第2種の実ページがあるときには、それを該第2種の仮想ページに割り当てる第20項の実記憶割り当て方法。

3. 発明の詳細な説明

〔発明の利用分野〕

本発明は仮想記憶方式の計算機システムにおいて各仮想ページに対する実記憶内の実ページの割り当て方法に係わり、特に大容量の実記憶装置を有する計算機システムに好適な実記憶割り当て方法に関する。

〔従来の技術〕

与のためのキー記憶装置の容量が巨大化し、しいては主記憶のアクセス性能が低下すると言う問題がある。

すなわち、記憶キー付与数の低減が必要である。

また、特開昭58-17586号でも、仮想ページサイズより実ページサイズが大きい場合の高速アドレス変換テーブルの構造上の工夫に関して論じられているが前記と同様の問題がある。

また、特開昭60-204048号では、仮想記憶および実記憶をあらかじめ複数の領域に分割し個々の領域では仮想ページサイズと実ページサイズが同一であるが、異なる領域間ではこのページのサイズが異なっている。実行するプログラムのメモリ利用特性に応じて各分割領域上のページサイズとページ数をあらかじめ決めておくことにより実記憶の有効活用とページの管理オーバーヘッドの削減を図っている。しかし、複数の分割された各実記憶領域毎においては個々の領域の実ページと同一サイズの仮想ページのみ割り当て可能であり各実記憶領域間の相互利用と言う点について

仮想ページに関する実ページの割り当てを管理する記憶管理方法に関しては、特開昭53-114318号、特開昭58-17586号、特開昭60-204048号、特開昭62-274351号記載の技術が知られている。

まず、特開昭53-114318号では記憶保のための記憶キーの付与単位(実ページに相当)2ⁿKB(キロバイト)の領域に対し仮想ページサイズが2ⁿKBで $m \geq n$ の範囲で可変であるような場合の高速アドレス変換に関して論じられており、高速アドレス変換テーブルエントリ長を拡張し2ⁿ個のキーを格納可能な構成にし、エントリ数を増やす事なくいずれの仮想ページサイズにおいても高速アドレス変換テーブル上に同数のページ数が登録可能としている。

この技術ではアドレス変換の効率に起因するシステムの性能については考慮されているが、記憶保護のための記憶キー設置単位による問題が考慮されておらず、キーの付与が一番小さい単位毎に必要であり、実記憶容量の巨大化に伴い記憶キー付

考慮がされておらず、あるサイズの仮想ページに実記憶を割り当てようとしても実記憶全体としては空き領域があるにもかかわらず仮想ページと同サイズの実ページを持つ実記憶領域に空き領域が無ければ割り当て不能となり、実記憶全体の効率的な利用と言う面で問題がある。すなわち、異なるサイズの仮想ページと実ページの相互割り当てがなされていない。

また、特開昭62-274351号では、実記憶上にセグメントサイズより大きな連続する空き領域があればセグメント単位に実記憶を割り当て、アドレス変換テーブル上に記憶しておき、このような場合にはアドレス変換の過程において実施される2つの変換単位、すなわちセグメント変換とページ変換の内、ページ変換を不要とすることによりアドレス変換の高速化、アドレス変換テーブルの容量削減を可能としている。

しかし、この技術はユーザプログラムからユーザの欲する任意のサイズで仮想記憶の確保、解放が成されると言う点について考慮されておらず、

たとえばセグメント単位で実記憶を割り当てたとしても、ユーザプログラムからその内部を部分的に解放されると仮想記憶、実記憶上に虫食い状態の使用不可領域が発生し利用効率が悪い、また、実記憶はセグメントより小さい単位の実ページに分割されて各実ページは互いに独立して任意の仮想ページに割り当てられるため連続した実記憶が空いてる確率は一般的に非常に稀であると言う問題がある。

本発明の目的は異種サイズの仮想ページと実ページ間で割り当てを可能とすることにより上記問題を解決する効率的な記憶管理方法を提供することにある。

〔問題点を解決するための手段〕

上記目的達成のため本発明の記憶管理方式では、仮想ページサイズと記憶保護のためのキーの付与単位である実ページサイズが異なる場合のページ割り当て方法として、実ページサイズに対して仮想ページサイズが $1/n$ 倍(n は正整数)の場合には、実ページ内を $1/n$ に分割管理し各々の分

n に分割し仮想ページサイズ単位で割り当てることにより、仮想ページサイズに依存せずキー付与単位である実ページサイズを大きくできる、したがって、実ページ数、キー記憶(第1図の5)のエントリ(第1図の6)の数を少なくでき、実記憶の増大にともなうキー記憶のエントリ数の巨大化を抑止できる。また、キー記憶エントリ数の巨大化抑止により、主記憶のアクセス性能低下を抑えることができる。

さらに、この場合は、ページの使用頻度を調べる場合、仮想ページサイズと実ページサイズの内大きい方のサイズ、すなわち、実ページサイズを単位としてページの使用頻度計測を実施することにより、計測ページ数を削減でき、ページ計測に費やすオペレーティングシステムの処理時間を短縮できる。

(2) 仮想ページが実ページサイズの n 倍の場合には、連続する n ケの実ページを仮想ページと同一のサイズを持つ実ページ1ケと考え一つ

割単位毎に仮想ページに割り当てる。

また、実ページサイズに対して仮想ページサイズが n 倍(n は正整数)の場合には仮想ページサイズの整数倍のアドレス有する実ページを先頭とする連続する n ケの実ページを渡し、この連続する n ケを仮想ページ1ページに対応させることにより仮想ページサイズと異なるページサイズから構成される実記憶領域を仮想ページサイズ単位に割り当てる。また、実ページサイズに対して仮想ページサイズが $1/n$ 倍および n 倍のものが混在する場合には、仮想ページと実ページのサイズの組み合わせにより、前記の $1/n$ 倍もしくは n 倍の場合のいずれかの方式でページを割り当てる。さらに、ページの使用頻度を調べる場合には、仮想ページサイズと実ページサイズの内大きい方のサイズを単位としてページの使用頻度計測を実施することにより前記目的を達成する。

〔作用〕

(1) 仮想ページサイズが実ページサイズより小さい場合($1/n$ 倍)、実ページ内部を $1/n$

の仮想ページに割り当てる。この場合は、仮想ページサイズを単位としてページの使用頻度計測を実施することにより、計測ページ数を削減でき、ページ計測に費やすオペレーティングシステムの処理時間を短縮できる。

(3) 仮想ページサイズが実ページサイズに対して $1/n$ 倍および n 倍のものが混在する場合には、仮想ページと実ページのサイズの組み合わせにより、上記の(1)もしくは(2)を適応することにより異なるサイズの仮想/実ページ間の相互利用が可能となり効率のよい実記憶の利用ができる。

〔実施例〕

以下、本発明の実施例を図面を参照して説明する。第1の実施例では、第1図に示すように仮想記憶11はサイズ4KB(キロバイト)の仮想ページ12の群により構成されている。実記憶10は実ページサイズの異なる2つの領域20、30に分割されている。領域20は4KBの実ページ21の群より成り、以下この領域4KBページ領

域と呼ぶ。領域30は64KBの実ページ31の群より成り、以下この領域30を64KBページ領域と呼ぶ。以下に、詳述するように本実施例では64KBページ領域30を構成する各64KBページ31を仮想ページサイズに等しい4KBの領域に分割し、各分割領域を1つの仮想記憶に割り当てて所に特徴がある。

5は実記憶10の各ページに対する記憶保護キーを保持するためのキー記憶である。その各エントリ6には対応する実ページに割り当てられた記憶保護キー(SK7)、そのページの参照の有無を示す参照ビット(R)8、そのページが書き替えられたことを示すチェンビット(C)9を格納するエリエから成る。キー記憶5の各エントリ6は4KBページ領域20に対して従来と同様にページごと、すなわち、4KBごとに、64KBページ領域30に対しても同様にページごと、すなわち、64KBごとに設置されている。すなわち、従来と同様に本実施例でも実ページは記憶保護キーの付与単位となっている。64KBページ領域

30では実ページサイズを4KBから64KBに拡張されているので、4KBページ領域20に比べ実記憶の単位容量当りのキー付与数を1/1.6に抑えキー記憶容量の増加を抑止している。

第2図は、オペレーティングシステムに含まれるプログラムおよび管理テーブルを示す。図において、2はそのうち実記憶割り当てプログラム、3はページ計測プログラム、7はページアウトプログラムを示す。実記憶割り当てプログラム2ではある仮想ページに対して、プログラムから実記憶の割り当て要求があると仮想ページ単位に実記憶の領域を割り当てる。この割り当てでは具体的には実記憶より4KBの空き領域を1つ選び出し割り当てるべき仮想ページに対応するアドレス変換テーブルのエントリ上に、選び出した実記憶領域のアドレスをセットし仮想アドレスから実アドレスへの変換を可能とする。すなわち、本発明でも従来と同様に仮想ページは仮アドレスから実アドレスへのアドレス変換単位である。ページ計測プログラム3は実記憶上に割り当てられている各

仮想ページの使用頻度を計測するもので、実記憶の不足状態に陥るとこの計測結果に従ってページアウトプログラム7で使用頻度の低い仮想ページに割り当てられている実記憶上の情報を順次、補助記憶にページアウトし使用可能な実記憶を一定量確保する制御に用いられる。図において、50、60は実記憶の現在の割り当て状況を管理するための実記憶管理テーブルである。4KB領域管理テーブル50は仮想ページサイズと同じサイズの実記憶領域、すなわち、4KB実ページの各々および各64KBページの各分割領域に対応するエントリ51を持つ。各エントリには対応する実記憶内の4KB領域の割り当て状況を示す情報を保持する。後に説明するようにこの管理テーブル50のエントリの内、仮想ページに未だ割り当てられていない4KBの実記憶領域に対応するエントリのキュー(未割り当て4KB領域キュー710(第4A図))が形成される。71はこのキューの先頭エントリを示すポインタである。また、64KB領域管理テーブル60は64KBページ

領域30内の64KBの実ページの各々に対応するエントリ61を持つ。各エントリは、対応する64KBページの割り当て状況を示す情報を保持する。後に説明するようにこの管理テーブル60のエントリうち、仮想ページに未だ割り当てられていない64KBの実記憶領域に対応するエントリのキュー(未割り当て64KB領域キュー720(第4B図))が形成される。72はこのキューの先頭エントリを示すポインタである。また、100はプログラム(ジョブ)ごとに設けたジョブ管理テーブルである。この管理テーブルには、後述するように、各ジョブが使用中の仮想ページに割り当てられた4KB実ページに対応するテーブル50内のエントリキュー(割り当て済み4KBページキュー810(第5A図))が形成される。81はこのキューの先頭エントリを示すポインタである。また、後述するように、各ジョブが使用中の仮想ページに割り当てられた分割領域に対応するテーブル50内のエントリキュー(割り当て済み分割領域キュー820(第5B図))が形

成される。82はこのキューの先頭エントリを示すポインタである。また、各ジョブについて各ジョブが使用中の64KB実ページの内、一部の分割領域が未割り当てであるものに対するテーブル60のエントリのキュー（部分未割り当て64KBページキュー910（第5C図））が形成される。91はこのキューの先頭エントリを示すポインタである。また、各ジョブについて、各ジョブが使用中の64KB実ページの内、その全ての分割領域が割り当て済であるものに対するテーブル60のエントリキュー（全割り当て済64KBページキュー920（第5D図））が形成される。92はこのキューの先頭エントリを示すポインタである。

第3A図は4KB領域管理テーブル50の1つのエントリ51内の情報を示す図である。このエントリ51は実記憶10の各4KBの領域毎に設けられている。フィールド91Aには実記憶のエントリに対応する4KBの領域に割り当てられた仮想ページの仮想アドレスがセットされ、フィー

ルド92Aにはその仮想ページを使用しているジョブの識別子が（ジョブID）がセットされる。フィールド93Aには、実記憶の対応する4KB領域が最近に使用されてから現在に至るまでの時間（未使用時間）がセットされている。なお、6KBの実ページ内を分割した4KB領域に対応するエントリ51にはセット不要である。フィールド94A、95Aには、そのエントリがテーブル50内の他のエントリとキューイングされるときに使用されるもので、フィールド94Aには、そのエントリの次につながるべきエントリのアドレス（フォワードポインタ（FWP1））がセットされ、フィールド95Aには、そのエントリの前つながっているエントリのアドレス（バックワードポインタ（BWP1））がセットされる。

第3B図は64KB領域管理テーブル60の1つのエントリ61内の情報を示す図である。このエントリ61は実記憶10の各64KBの実ページに対応して設けられる。フィールド91B、92B、93B、94B、95Bはそれぞれの第

3A図のフィールド91A、92A、93A、94A、95Aと同じ意味の情報がセットされる。フィールド96には、そのエントリに対応する64KBページ内の16個の分割領域の内、仮想ページに未割り当ての分割領域数がセットされる。フィールド97には、対応する64KB実ページ内の分割領域が最初に使用された時に分割領域に割り当てた仮想ページの持つ記憶保護キーがセットされる。この記憶キーは後述するように64KB実ページ内に割り当てる16個の仮想ページの記憶キーが矛盾しないで同一となるよう制御に使用する。フィールド97、98はそのエントリに対応する64KBページ内の16個の分割領域に対する4KB領域管理テーブル50内のエントリ51のキューを形成するときに使用される。

第4A図は4KB未割当領域キュー710を示す。このキューは実記憶10の各4KB領域の内、仮想ページが割り当てられていない領域に対するテーブル50のエントリ51Aの群をポインタにより連ねて構成される。実記憶の4KBの未割当

領域に対する各エントリ51A内のフォワードポインタ（FWP1）は後続の未割当の4KB領域に対応するエントリ51Aのアドレスがセットされ、そのエントリ51A内のバックワードポインタ（BWP1）には、先行する未割当の4KB領域に対する他のエントリ51Aのアドレスがセットされている。さらに4KB未割当領域キューポインタ内にはこのエントリ列の内、先頭にあるエントリのアドレスおよび末尾のエントリのアドレスをそれぞれフォワードポインタ71A、バックワードポインタ71Bとしてセットされている。こうしてキュー710が形成される。なお、このようなエントリ51Aがない場合にはポインタ71A、71Bには0がセットされる。このことは他のキューについても同様である。

ポインタ71につらなるエントリ51Aの列をたどることにより、未割当の4KB領域がいずれであるかを判定できる。

第4B図は、64KB未割当ページキュー720を示す。このキューは、実記憶10の64KB

ページの内、どの分割領域も仮想ページに割当てられていないものに対する、テーブル60のエントリ61Bの列およびそれぞれの64KBページ内の16個の分割領域に対するテーブル50のエントリ51Bの列からなる。すなわち未割当の64KBページに対するテーブル60のエントリ61Bのフォワードポイント(FWP2)、バックワードポイント(BWP2)には、それにつづき、未割当の64KBページに対するエントリ61Bのアドレスおよびそれに先行する未割当の64KBページに対するエントリ61Bのアドレスがそれぞれセットされる。エントリ51Bの列の先頭のエントリおよび末尾のエントリのアドレスがそれぞれ、キューポイント72内に、フォワードポイント、バックワードポイントとしてセットされる。こうしてキュー720が形成される。さらに、未割当の64KBページに含まれる16個の分割領域に対するテーブル50のエントリ51Bのフォワードポイント(FWP1)とバックワードポイント(BWP1)には、同じ64KBペ

域キュー820を示す。

このキュー820は、そのジョブに割当て済の分夜領域に対するテーブル50のエントリ51Dの列をキューポイント82とからなる。第5C図はあるジョブに対する部分割り当て済み64KBページキュー910を示す。

このキューはそのジョブに一部の分割領域が割当て済みであるが、他の一部の分割領域が未割当てである64KBページに対するテーブル60のエントリ61Eの列をそれぞれのページ内の未割当の分割領域に対するテーブル50のエントリ51Eの列とキューポイント91とから形成される。これらのエントリ61E、51Eの接続は未割当64KBページキュー810(第4B図)の場合と同様である。

なお、ジョブ管理テーブル100内の、64KB部分使用ページキューポイント102には、これらのエントリ61の列の先頭のエントリおよび末尾のエントリのアドレスがそれぞれフォワードポイント102A、バックワードポイント102

ージ内の、後続の分割領域に対する、テーブル50のエントリ51Bのアドレスおよび同じ64KBページ内の先行する分割領域に対する、テーブル50のエントリ51Bのアドレスがそれぞれセットされることにより、同じ64KBページに属するエントリ51Bの列が形成される。この64KBページに対する、テーブル60内のエントリ61B内の内部フォワードポイント(F)および内部バックワードポイント(L)には、そのようなエントリ51の列の先頭のエントリ51Bおよび末尾のエントリ51Bのアドレスがセットされている。

第5A図はあるジョブに対する割当済4KBページキュー810に示す。キュー810はそのジョブに割当済みの4KB実ページに対するテーブル50のエントリ51Cの列とキューポイント81にて形成される。これらのエントリの接続は未割当4KB領域キュー710(第4A図)の場合と同じである。

第5B図は、あるジョブに対する割当済分割領

Bとしてセットされる。このキュー910からあるジョブが使用している64KBページの内、未割当ての分割領域を容易に検出できる。

第5A図～第5C図に示したキュー810、820、910は他のジョブに対しても同様に設けられている。それらは簡単化のために図示されていない。

第5D図は、あるジョブに対する全割り当て済み64KBページ、920を示す。このキューは、そのジョブに割り当てられている、64KBページのうち、その分割領域が全て割り当て済みである(全てが使用済みである)64KBページを対象に、それに対応するテーブル60のエントリ61Fの列から形成される。このキュー920は、後で述べるページ計測プログラム3により使用される。また、このキュー920は、前記部分割り当て済みキュー310とは異なり、全ての分割領域が使用済みであるため、それら分割領域に対するテーブル50のエントリ列は接続されない。これら分割領域に対するエントリは、前記の通り割

り当て済み分割領域キュー820に接続されている。

次に実記憶割り当てプログラム2の動作を説明する。

第6図はページ割り当てプログラム2の概略フローを示す図である。

あるプログラム(ジョブ)の実行中に仮想記憶上のある領域をアクセスしたとき、その仮想領域に対応する実記憶領域が割り当てられていない場合に、ページフォルト割り込みが発生し、これを契機に仮想領域に実記憶領域を仮想ページ単位に割り当てるための実記憶割り当て要求がおこり、実記憶割り当てプログラム2が起動される。実記憶割り当てプログラム2が起動されると、まず実記憶領域選定ステップ200で割り当てるべき実記憶上の領域を選定する。

本実施例では割り当てるべき実記憶が属する領域が4KBページ領域20又は、64KBページ領域30のいずれかにあらかじめ指定されている場合にはその指定領域を選定し、その他の場合には、

まず64KBページ領域30を選定する。後に説明するようにこの領域30内の未割り当ての4KB領域を捜し、これが見つからない場合には4KBページ領域20内の実記憶を割り当てるようになっている。実記憶領域選定ステップ200により64KBページ領域30が選ばれると、要求元ジョブ割り当て済み64KBページ内サーチ210を実施する。この処理は64KBページ内を分割使用した場合に生じる64KBページ内の部分未割り当て分割領域を利用するための工夫であり、要求元ジョブに概に部分的に割り当て済みの64KBページ内の未割り当て分割領域を優先してそのジョブに割り当てる。

ジョブからの実記憶割り当ての要求がそのジョブから最初に出された場合、そのジョブに対して部分的に未割り当ての64KBページは存在しない。従って、サーチ210は失敗するのでこのサーチ210については後に説明する。サーチ210で部分的に未割り当てのページが見つからなければ、次に未割り当て64KBページサーチ230を実

施し、未割り当て64KB内の一つの分割領域を割り当てる。すなわち、未割り当て64KBページキュー720(第4B図)を用いて、そのジョブに割り当てるべき未使用の64KBページを見つける。このために、まずステップ231でこのキュー720のポインタ72を取り出し、ステップ232でポインタ値のゼロ判定を行いこのポインタに接続されたエントリ61Bの有無を調べる。

キューポインタ72がゼロでなければ、ステップ233にてそれに連なるキュー720の先頭のエントリ61Bを取り出す。そのエントリ61Bに連なる16個の分割領域に対するエントリ51Bの列の先頭のエントリを選択しそれに対応する分割領域を割り当て対象とする。次に、ステップ234Aで管理テーブルエントリおよびキューの更新を行う。すなわち、エントリ61Bの内部フォワードポインタ(F)(第3B図)で示されるエントリ61Bに連なるエントリ51Bを選択し、そのフィールド91A, 92A, (第3図A)に割り当て対象となった仮想ページのアドレスと割

り当て要求元のジョブのIDをセットし、選択したエントリ61B内の割り当て済み分割領域数96を+1し(その初期値は0である)、フィールド92BにそのジョブのジョブIDをセットする。

次に、キューの更新を以下のように行う。即ち、未割り当て64KBページキュー720から(第4B図)上記選択されたエントリ61Bおよびこれに連なる16個のエントリ51Bをはずし、このエントリ61Bおよび上記16個のエントリ51Bの中の、上記選択されたエントリ51B以外の15個(未割り当ての分割領域に対するテーブル50のエントリ)をそのジョブに対して設けられた部分割り当て済み64KBページキュー910(第5C図)に接続する。さらに、上記選択されたエントリ51Bをそのジョブに対して設けられた割り当て済み分割領域キュー820(第5B図)に接続する。

次に、ステップ236Aにて割り当て対象となった仮想ページに指定されている記憶キー(ジョ

ブから明示的に指定されていない場合は、そのジョブのその時点のプログラムキー値)をフィールド97にセットし、割り当て対象となった分割領域の含まれる64KB実ページに対応するキー記憶5(第1図)のエントリ6にフィールド97にセットした値と同じキー値をセットする。なお、この64KB実ページのアドレスは選択されたエントリ61Bがテーブル60中の何番目のエントリであるか(すなわち、それに対応して定まるこのエントリ61Bのテーブルインデックスがそのまま実ページのページアドレスをページ番号で示す)から決定することができる。この64KB実ページアドレスに基づいてハードウェア命令を用いて上記記憶キーを書き込む。また、この記憶キーの書き込みは未割り当て64KBページ内から分割領域を最初に割り当てる場合のみ実施するのは前述の通りである。

次に、ステップ240により、実記憶割り当てが要求された仮想ページに対応するアドレス変換テーブル内のエントリに選択した分割領域の実アド

レスのセットとこのエントリのバリデーションを実施する。こうしてこの実記憶領域が使用可能となる。こうして、未割り当ての64KBページ内の最初の分割領域の割り当てが終了する。

以上の処理における、サーチ230においてキューポインタ(72)がゼロであれば未割り当ての64KBページは無しと判断し4KBページの選択処理260にて実記憶10の4KB実ページ領域20内の未割り当ての一つの実ページを割り当てるべき実記憶領域として選択する。

この選択は、未割り当て4KBページキュー710(第4A図)に接続された一つのエントリ51Aを選択することにより行われる。次にステップ234Bにおいて、この選択されたエントリ51A内に割り当て要求元のジョブのIDおよび割り当て対象の仮想ページのアドレスを書き込む。さらに、このエントリ51Aを上記キュー710よりはずし、そのジョブに対して設けられた割り当て済4KBページキュー810(第5A図)に接続する。その後、ステップ236Bにおいて記

憶キーのセットをステップ236Aと同様に行い、ステップ240においてアドレス変換テーブルの書き換えを先に述べたのと同じように行う。こうして4KB実ページへの割り当てが終了する。その後、仮想ページに対する新たな実記憶割り当て要求がなされる毎に同じように処理される。この際、先に詳細を省略した要求元ジョブ割り当て済み64KBページサーチ210も実行される。以下、その詳細を述べる。

すなわち、ステップ211においてそのジョブに対する部分未割り当て64KBページキュー910(第5C図)のキューポインタ91を読み出し、ステップ212でその値がゼロか否かを判断しポインタ91の値がゼロであれば部分未割り当ての64KBページは無いと判断し、前述の未割り当て64KBページサーチ230に移る。

ポインタ91がゼロでなければ、ステップ213でこのキューポインタ91に接続された先頭のエントリ61Eを選択し、次にステップ220でその先頭のエントリ61Eの記憶キー97(第3

B図)と割り当て対象仮想ページの記憶キーとを比較し、一致すれば割り当て可能と判断する。これが不一致であればステップ221でその先頭エントリ61E内のフォワードポインタFWP2(第5C図)を取り出しこれを新たなポインタとして、ステップ212から221を後続のエントリ61E内のフォワードポインタFWP2がゼロとなるか、前記ステップ220で記憶キーが一致するエントリ61Eが見つかるまで繰り返す。このページキュー910にて示される64KBページはいずれも、今回のページ割り当て要求元ジョブに一部の部分領域が割り当て済みである。今回のページ割り当て対象仮想ページの記憶キーと一致する記憶キーを持つ64B実ページがその部分割り当て済みページキュー中に有れば、その部分割り当て済み64Bページ内の一つの分割領域を割り当てるべき実記憶領域として選択する。このようにすれば64KB実ページに対して1つのキー記憶エントリしか与えられない状態においても、そのページを分割領域に分けてジョブに割り当て

ても不都合が生じない。さて、上記のようにしてステップ220において部分割り当て済み64KBページキュー910の一つのエントリ61E内の記憶キーが一致する場合には、ステップ234Cにてそのエントリおよびキューの更新を行う。すなわち、そのエントリ61Eに接続されていた未使用分割領域のエントリ51Eを一つ選択し、このエントリに対応する分割領域を割り当て領域とすべく、この選択されたエントリ51EにそのジョブIDおよび割り当て対象となった仮想ページのアドレスをセットする。さらに、上記エントリ61E内の割り当て済み分割領域数96を+1する。

上記エントリ51Eを上記キュー910からはずし、そのジョブに対して設けられている割り当て済み分割領域キュー820に接続する。もし、上記更新後の割り当て済み分割領域数が16に等しいときは、割り当て対象となった分割領域を含む64KB実ページは未割り当ての分割領域がなくなったことになる。したがって、この時には、

ページ計測の目的について少し述べておく。ページ計測とは、実記憶不足時にページアウトすべきページを決定するために、実行中の個々のジョブが一定のCPUを使用するつど各ジョブに割り当てられているページの使用頻度を前もって調べておく処理である。したがって、本ページ計測プログラム3はある時間インターバルで起動される。

本実施例におけるページ計測処理は、仮想ページとそれに割り当てられた実記憶が属する実ページサイズが異なる場合、その仮想ページと実ページのサイズの大きい方の領域毎にページ計測を行う所にある。

すなわち、64KBページがいくつかの仮想ページに割り当てられている場合、その実ページの16個の分割領域のいずれかが使用された場合、その実ページが使用されたものとしてページ計測を行う。すなわち、64KBページ領域30上に割り当てられている4KB仮想ページに対しては、割り当て済み分割領域キュー820（第5B図）を使用せず、部分割り当て済み64KBページキ

この64KB実ページに対応する前記エントリ61Eもキュー910からははずし、全割り当て済みキュー920（第5D図）にエントリ61Fとして接続する。

こうして、ステップ234Cが終了するとその後ステップ240が前と同様になされる。以上の処理を実ページ割り当て要求がなされるたびに繰り返せばよい。こうして、仮想ページサイズは従来通りの4KBのまま、実ページサイズのみを拡張した場合についても、キー記憶5（第1図）上のエントリ6を64KBの実ページに対して1つしか設けずに、かつ実記憶に空きを生じないで実記憶領域を仮想ページに割り当てることができる。

次にページ計測プログラム3（第2図）の動作を説明する。第7A、7B図はページ計測プログラム3の概略フローを示す図である。

点線で示したステップ370～380は本ケースの実施例では使用せず、後で述べる第2の実施例で用いる。

ページ計測プログラム3の動作説明の前にペー

キュー910（第5C図）、および全割り当て済み64KBページキュー920（第5D図）を用いてこれらキューに接続されているテーブル60のエントリ61単位に、すなわち、64KB単位にページ計測を実施する。

以下、詳細を説明する。第7A図に示すように、ステップ310でページ計測すべきジョブに対して設けられた全割り当て済み64KBページキュー920（第5D図）の先頭ポインタ92を変数QUEUEにセットし、ステップ320で内部サブルーチンSUB1をコールする。

このキュー920は4KBの仮想ページが割り当てられている64KB実ページに対するテーブル60のエントリを表すキューの一つで前述の64KB実ページ内の全分割領域が割り当て済みの64KBページを表す。サブルーチンSUB1はコールされると、ステップ510において、変数QUEUEの値がゼロか否かによりページ計測すべきエントリ61Fがあるか否かを調べる。次にエントリがある場合にはステップ440によ

り変数QUEUEで指されるエントリ61Fに対応する64KB実ページの使用の有無を調べる。これは、ハードウェア命令により第1図に示すキー記憶エントリ6上の参照ビット(R7)を読み出すことにより行う。参照ビットR7は当該キー記憶エントリに対応するページがアクセスされるとハードウェアによりセットされ、前記ハードウェア命令により命令実行後リセットされる。これらのビットがゼロであれば当該ページは、前回のページ計測時点から今回のページ計測までの間使用されなかったことが分かる。次に、当該ページが使用されていれば、ステップ450において不参照期間数93A(第3B図)をクリアし(不参照期間数93Aをゼロにする)、ステップ460で当該エントリ61Fを当該キュー(SUB1がコールされたとき変数QUEUEにキューポインタをセットしたキュー、この場合はキュー920(第5D図))の最終位置に移動する。このエントリの移動は、不参照期間数93Aの大きい順にキューを構成するためである。こうすることによ

ページキュー920(第5D図)に接続されているものについて64KB単位にページ計測を実施した。

次に、64KBページ領域30上に割り当てられている4KB仮想ページの内、もう一つの部分割り当て済み64KBページキュー910(第5C図)に接続されているものについて、64KB単位にページ計測を実施するため、第7A図に示すように、ステップ330で変数QUEUEこのキューの先頭エントリ61Eを示すポインタ91をセットしステップ340で内部サブルーチンSUB1をコールする。サブルーチンSUB1は、前記キュー910に接続されているエントリ61Eを対象として先ほど説明したステップ510~480の処理を繰り返すことにより当該キュー上に接続されるエントリ61Eに対するページ計測処理を64KB単位に行う。以上で64KBページ領域30上に割り当てられている、当該ジョブのページ計測が終了する。次に、4KBページ領域20上に割り当てられている4KB仮想ページ

り、後に述べるページアウトプログラム7では、キューの先頭から順次ページアウトすれば自動的に使用頻度の小さいものよりページアウトできる。次に、当該ページが使用されていなければステップ470において不参照期間数をカウントアップする。このように、ページの使用頻度はページ計測インターバル毎にそのページが使用されていない場合に不参照期間数をカウントアップし、使用されていた場合には不参照期間数をゼロにするため、この不参照期間数が大きいほど使用頻度が低いことを示す。

次にステップ480により変数QUEUEに現在処理しているエントリ61F内のフォワードポインタ(FWP2)94Bをセットし、新たなエントリ61Fに対して今まで述べてきたステップ510~480の処理を変数QUEUEの値がゼロすなわちキュー920に接続された全てのエントリ61Fを処理し終わるまで繰り返す。以上で、64KBページ領域30上に割り当てられている4KB仮想ページの内、全割り当て済み64KB

に対してページ計測を実施する。この場合には、ページ計測処理は4KB単位に実施する。4KBページ領域20上に割り当てられている4KB仮想ページは全て、割り当て済み4KBページキュー810(第5A図)に接続されている。この場合も、前記同様にステップ390で変数QUEUEにこのキューの先頭エントリ51Cを示すポインタ81をセットしステップ400で内部サブルーチンSUB1をコールする。サブルーチンSUB1は、前記キュー810に接続されているエントリ51Cを対象として前記同様にステップ510~480の処理を繰り返すことにより当該キュー上に接続されるエントリ51Cに対するページ計測処理を4KB単位に行う。以上の処理を実行中のジョブがCPUをある一定時間使用するつど実施すればよい。こうして、仮想ページサイズは従来通りの4KBのままで、ページ計測処理はその仮想ページに割り当てられた実記憶の実ページサイズ単位に実施することができ、ページ計測数を低減することができる。

次にページアウトプログラム7 (第2図) の動作を説明する。第13A, 13B図はページアウトプログラム7の概略フローを示す図である。本実施例におけるページアウト処理は、仮想ページとそれに割り当てられた実記憶が属する実ページサイズが異なる場合、その仮想ページと実ページのサイズの大きいほうのサイズ単位ページアウトを行う所にある。すなわち、64KBページ領域30内の同一64KB実ページ上の16個の分割領域に割り当てられている仮想ページ群を一括してページアウト処理する。こうすることによりI/O回線を低減することができる。以下、第13A図, 13B図により、ページアウト処理の詳細を述べる。本ページアウトプログラム7は実記憶の使用可能な領域が、ある一定量以下になると起動され、また、不参照期間数があらかじめ定められたn以上のページをNPAGE個ページアウトするべく起動されるものとする。ページアウトプログラム7は起動されると、まず、ステップ500でページアウトしたページ数をカウントする変

数、POUTを0にする。次に、ステップ510でページアウトすべきジョブに対して設けられた部分割り当て済み64KBページキュー910

(第5B図)の先頭ポインタ91を変数RPNにセットし、ステップ511で64KBページ領域30上に割り当てられているページのキューであることを示すために変数EXに1をセットし、ステップ512で内部サブルーチンOUTをコールする。サブルーチンOUTはコールされると、ステップ600において、変数RPNの値がゼロか否かによりページアウトすべきエントリ61Eがあるか否かを調べ、エントリが無ければ、何もせずリターンする。次にエントリがある場合にはステップ610により当該エントリ61Eの不参照期間数がページアウトプログラム起動時に指示されたn以上かを調べる。n以下であれば、ページ計測プログラム3で述べたように各エントリは不参照期間数が大きい順に並んでいるため以降のエントリ61Eもn以下であり、当該キューには今回のページアウト起動条件を満たすページが無

いためリターンする。

n以上であれば、次にステップ620でページアウトして良いページかどうかを調べる。これは、そのページのページアウトが禁止されている(ページ固定されている)とか、既に前回のページアウト処理でページアウト中であると言った従来からのチェックで、仮想記憶方式の計算機において通常成されているチェックである。ページアウト不可であれば、キューに接続された次のページを処理するため、ステップ680で変数RPNに当該エントリ61Eのフォワードポインタ94Bをセットしステップ600からの処理を繰り返す。ステップ620でページアウト可能なページが見つければ、次にステップ630で変数EXの値を調べる。この場合はEXに1がセットされており64KB単位の処理であると判断し、ステップ640に進む。ステップ640では当該エントリ61E上に含まれる16個の分割領域を全てページアウトし(但し当該エントリ61Eの対応するキー記憶上のチェンジビットC(第8図5)が

offの場合はページアウトは行わない)、各分割領域に割り当てられていた仮想ページに対応するアドレス変換テーブルのエントリを無効とする。次に、ステップ650で前記各分割領域に対応するエントリ51Eを部分割り当て済み64KBページキュー910から全てはずし、当該エントリ61E上のキューポインタ98, 99(第3B図)で管理される64KBページ内部のキューへ接続し、割り当て済み分割領域数96(第3B図)を0にセットし、さらに、エントリ61Eを未割り当て64KBページキュー720(第4B図)に接続する。次に、ステップ660でページアウト数を示す変数POUTを16増し、ステップ670でPOUTがページアウトすべき個数に達したか否かをチェックするため変数NPAGEと比較する。ページアウト数がNPAGE未満であれば、つぎにステップ680で変数RPNに当該エントリ61Eのフォワードポインタ94Bをセットしステップ600からの処理を繰り返す。以上の処理をステップ600で当該キューの全エントリ

61Eを処理し終わるか、ステップ610で起動時の条件以上の不参照期間数を持つエントリ61Eが現れるまで、もしくは、ステップ670で所定のページ数(NPAGE)をページアウトするまで繰り返す。以上で、64KBページ領域30上に割り当てられている4KB仮想ページの内、部分割り当て済み64KBページキュー910(第5C図)に接続されているものについて64KB単位にページアウトを実施しサブルーチンOUTがリターンする。次に、第13A図のステップ520で所定のページ数(NPAGE)をページアウト出来たかどうかを調べ、所定数に達していなければ、次に、64KBページ領域30上に割り当てられている4KB仮想ページの内、もう1つの全割り当て済み64KBページキュー920(第5D図)に接続されているものについて、64KB単位にページアウトを実施する。これは、第13A図に示すように、ステップ530で変数RPNこのキューの先頭エントリ61Fを示すポインタ92をセットしステップ531で64KB

ページ領域30上に割り当てられているページのキューであることを示すために変数EXに1をセットしステップ532で内部サブルーチンOUTをコールする。

サブルーチンOUTはコールされると今度は、キュー920上のエントリ61Fに対して前記と同様に先ほどまで説明したステップ600~680の処理を繰り返すことにより当該キュー920上に接続されるエントリ61Fに対するページアウト処理を64KB単位に行う。但し、ステップ650でキューの更新する際、エントリ61Fをキュー920から外す処理が追加されることが異なる。以上で64KBページ領域30上に割り当てられている、当該キューのページアウトが終了しサブルーチンOUTはリターンする。次に、再度第13A図のステップ540で所定のページ数(NPAGE)をページアウト出来たかどうかを調べ、所定数に達していなければ、次に、4KBページ領域20上に割り当てられている4KB仮想ページに対してページアウトを実施する。

この場合には、ページアウト処理は4KB単位に実施する。4KBページ領域20上に割り当てられている4KB仮想ページは全て、割り当て済み4KBページキュー810(第5A図)に接続されている。この場合も、前記同様にステップ550で変数RPNにこのキューの先頭エントリ51Cを示すポインタ81をセットしステップ551で、4KBページ領域20上のキューであることを示すためEXに0をセットし、ステップ552で内部サブルーチンOUTをコールする。サブルーチンOUTは、前記キュー810に接続されているエントリ51Cを対象として前記同様にステップ600~630のまでの処理を行うと、次にステップ630で変数EXの値を調べる。この場合はEXに0がセットされており4KB単位の処理であると判断し、ステップ640Aに進む。ステップ640Aでは当該エントリ51Cに対応するページをページアウトし(但し当該エントリ51Cの対応するキー記憶上のチェンジビットC(第8図5)がoffの場合はページアウトは行

わない)、当該領域に割り当てられていた仮想ページに対応するアドレス変換テーブルのエントリを無効とする。次に、ステップ650Aで前記各エントリ51Cを割り当て済み4KBページキュー810からはずし、当該エントリ51Cを未割り当て4KBページキュー710(第4A図)に接続する。次に、ステップ660Aでページアウト数を示す変数POUTを1増し、ステップ670でPOUTがページアウトすべき個数に達したかどうかをチェックするため変数NPAGEと比較する。ページアウト数がNPAGE未満であれば、つぎにステップ680で変数RPNに当該エントリ51Cのフォワードポインタ94A(第3A図)をセットしステップ600からの処理を繰り返す。以上に処理をステップ600で当該キューの全エントリ51Cを処理し終わるか、ステップ610で起動時の条件以下の不参照期間数を持つエントリ51Cが現れるまで、もしくは、ステップ670で所定のページ数(NPAGE)をページアウトするまで繰り返す。

以上で、4 KB ページ領域 20 上に割り当てられている割り当て済み 4 KB ページキュー 810 (第 5 A 図) に接続されているものについて 4 KB 単位にページアウトを実施しサブルーチン OUT がリターンし、処理を終える。以上の様にして十分な未使用実記憶量が確保出来るまで、ページアウトプログラム 7 起動時の条件の 1 つであるページアウトすべき参照期間数を順次小さくして繰り返しページアウトプログラム 7 を起動すれば良い。こうして、仮想ページサイズは従来通りの 4 KB のままで、ページアウト処理はその仮想ページに割り当てられた実記憶の実ページサイズ単位に実施することができ、ページアウトに関する I/O 回数を低減できる。

以上で本発明の第 1 の実施例の説明を終える。

次に、本発明の第 2 の実施例について述べる。

なお、第 2 の実施例の説明に当り第 1 実施例と同じ構成、用途のものは第 1 実施例と同じ番号を付加し、その詳細図も同一の図を用いる。

領域 30 は 64 KB の実ページ 31 の群より成り、以下この領域 30 を 64 KB ページ共用領域と呼ぶ。

領域 30 A は 64 KB の実ページ 31 A の群より成り、以下この領域 30 A を 64 KB ページ専用領域と呼ぶ。なお、この両者をまとめて 64 KB ページ領域と呼ぶ。すなわち、本実施例では第 1 の実施例に領域 20 A と領域 30 A が追加されている所が異なる。領域 20 は第 1 実施例で詳述したように本実施例においても 4 KB 仮想ページのみを割り当て可能な領域である。領域 30 も第 1 実施例と同様に 64 KB ページ領域 30 構成する各 64 KB ページ 31 を 4 KB の領域に分割し、各分割領域 1 つの 4 KB 仮想ページに割り当てることも出来るし、各 64 KB ページ 31 を 1 つの 64 KB 仮想ページに割り当てることも出来る所が特徴の一つである。また、領域 20 A は領域を構成する各 4 KB ページ 21 A を 1 つの 4 KB 仮想ページに割り当てることも出来るし、16 個の連続する 4 KB ページ 21 A をまとめて 1 つの

第 2 の実施例では、第 8 図に示すように仮想記憶 11 A は仮想アドレス

2 GB (ギガバイト) を境としてサイズ 4 KB (キロバイト) の仮想ペ

ージ 12 の群により構成されている 4 KB ページ領域 110 とサイズ 64 KB (キロバイト) の仮想ページ 13 の群により構成されている 64 KB ページ領域 120 から成る。

実記憶 10 A は第 1 の実施例と同様に実ページサイズの異なる 2 つの領域、すなわち、ページサイズ 4 KB 領域と 64 KB 領域に分割されている。さらに、本実施例では各ページサイズの領域内がさらに各々 2 分割されており、合計 4 つの領域 20, 20 A, 30, 30 A に分割されている。

領域 20 は 4 KB の実ページ 21 の群より成り、以下この領域を 4 KB ページ専用領域と呼ぶ。

領域 20 A は 4 KB の実ページ 21 A の群より成り、以下この領域を 4 KB ページ共用領域と呼ぶ。なお、この両者をまとめて 4 KB ページ領域と呼ぶ。

64 KB 仮想ページに割り当てることも出来る所が特徴の一つである。

また、領域 30 A は 64 KB 仮想ページのみが割り当て可能な領域である。

5 は実記憶 10 A の各ページに対する記憶保護キーを保持するためのキー記憶である。その各くエントリ 6 には対応する実ページに割り当てられた記憶保護キー (SK7)、そのページの参照の有無を示す参照ビット (R) 8、そのページが書き替えられたことを示すチェンビット (C) 9 を格納するエリアから成る。キー記憶 5 の各エントリ 6 は 4 KB ページ領域 20, 20 A に対しては従来と同様にページごと、すなわち、4 KB ごとに、64 KB ページ領域 30, 30 A に対しても同様にページごと、すなわち、64 KB ごとに設置されている。すなわち、従来と同様に本実施例でも実ページは記憶保護キーの付与単位となっている。64 KB ページ領域 30, 30 A では実ページサイズを 4 KB から 64 KB に拡張されているので、第 1 の実施例と同様に 4 KB ページ領域 20,

20Aに比べ実記憶の単位容量当りのキー付与数を1/16に抑えキー記憶容量の増加を抑止している。

第9図は、オペレーティングシステムに含まれるプログラムおよび管理テーブルを示す。図において、2Aはそのうちの実記憶割り当てプログラム、3はページ計測プログラム、7Aはページアウトプログラムを示す。実記憶割り当てプログラム2Aではある仮想ページに対して、プログラムから実記憶の割り当て要求があると仮想ページ単位に実記憶の領域を割り当てる。この割り当てでは具体的には実記憶より仮想ページサイズに一致する空き領域を1つ選び出し割り当てるべき仮想ページに対応するアドレス変換テーブルのエントリ上に、選び出した実記憶領域のアドレスをセットし仮想アドレスから実アドレスへの変換を可能とする。すなわち、本発明でも従来と同様に仮想ページは仮想アドレスから実アドレスへのアドレス変換単位である。ページ計測プログラム3は実記憶上に割り当てられている各仮想ページの使用

頻度を計測するもので、実記憶の不足状態に陥るとこの計測結果に従ってページアウトプログラム7Aで使用頻度の低い仮想ページに割り当てられている実記憶上の情報を順次、補助記憶にページアウトし使用可能な実記憶を一定量確保する制御に用いられる。図において、50、60は実記憶の現在の割り当て状況を管理するための実記憶管理テーブルである。4KB領域管理テーブル50は仮想ページサイズと同じサイズの実記憶領域、すなわち、4KB実ページの各々(21, 21A群)および各64KBページの各分割領域(31群)に対応するエントリ51を持つ。各エントリには対応する実記憶内の4KB領域の割り当て状況を示す情報を保持する。後に説明するようにこの管理テーブル50のエントリの内、仮想ページに未だ割り当てられていない4KBの実記憶領域に対応するエントリのキュー(未割り当て4KB領域キュー710(第4A図))が形成される。71はこのキューの先頭エントリを示すポインタである。

また、64KB領域管理テーブル60は4KBページ領域20Aの連続する4KB実ページ21Aを16個毎にまとめた64KB単位に対応するエントリ61、および、64KBページ領域30内の64KBの実ページ31の各々に対応するエントリ61を持つ。各エントリは、対応する実記憶領域の割り当て状況を64KB単位に示す情報を保持する。

後に説明するようにこの管理テーブル60のエントリの内、仮想ページに未だ割り当てられていない64KB単位の実記憶領域に対応するエントリのキュー(未割り当て4KB共用キュー1720(第10A図)、部分割り当て済み4KB共用キュー1730(第10B図)、未割り当て64KB領域キュー720(第4B図))が形成される。172, 173, 72はこれらキューの先頭エントリを示すポインタである。

また、64KB領域管理テーブル40は64KBページ領域30A内の64KBの実ページ31Aの各々に対応するエントリ41を持つ。各エント

リは、対応する実記憶領域の割り当て状況を64KB単位に示す情報を保持する。

後に説明するようにこの管理テーブル40のエントリうち、仮想ページに未だ割り当てられていない64KB単位の実記憶領域に対応するエントリのキュー(未割り当て64KB専用キュー1750(第10C図))が形成される。

175はこのキューの先頭エントリを示すポインタである。

また、100Aはプログラム(ジョブ)ごとに設けたジョブ管理テーブルである。この管理テーブルには、後述するように、各ジョブが使用中の仮想ページに割り当てられた4KB実ページに対応するテーブル50内のエントリキュー(割り当て済み4KBページキュー810(第5A図))が形成される。

81はこのキューの先頭エントリを示すポインタである。

また、後述するように、各ジョブが使用中の仮想ページに割り当てられた分割領域に対応するテ

ーブル50内のエントリキュー（割り当て済み分割領域キュー820（第5B図））が形成される。82はこのキューの先頭エントリを示すポイントである。また、各ジョブについて各ジョブが4KBの仮想ページ割り当てに使用中の64KB実ページの内、一部の分割領域が未割り当てであるものに対するテーブル60のエントリのキュー（部分未割り当て64KBページキュー910（第5C図））が形成される。91はこのキューの先頭エントリを示すポイントである。また、各ジョブについて各ジョブが4KBの仮想ページ割り当てに使用中の64KB実ページの内、全部の分割領域が割り当て済みであるものに対するテーブル60のエントリのキュー（全割り当て済み64KBページキュー920（第5D図））が形成される。92はこのキューの先頭エントリを示すポイントである。

また、各ジョブが使用中の64KB仮想ページが割り当てられている64KB実ページ31Aに対するテーブル40のエントリ41のキュー（割り

ページ領域20A内の領域を64KB単位で管理する。仮想ページが割り当てられていない連続する16個の4KB実ページ21A群毎に設けられたテーブル60のエントリ61Gの列および前記4KB実ページ21A群の各々に対するエントリ51Gの列とキューポイント172とから形成される。各エントリのキューのされ方は前記キュー720と同様であり省略する。

第10B図は部分割り当て済み4KBページ共用キュー1730を示す。このキューは第1実施例のキュー910（第4C図）と同形式であるが、第1実施例とは異なりジョブ別に作成されずシステム全体で唯一個のキューを形成する。

このキューは実記憶10Aの4KBページ領域20A内の領域を連続する16個の4KB実ページ21A群毎の64KB領域の内、前記連続する16個の4KB実ページ21A群のいずれかが4KB仮想ページに割り当てられているテーブル60のエントリ61Hの列および前記4KB実ページ21A群の各々に対するエントリ51Hの列と

当て済み64KBページキュー1760（第11図））が形成される。176はこのキューの先頭エントリを示すポイントである。

第3A図、第3B図に各管理テーブルエントリ51、エントリ61の構造を示すが、第1実施例ですでに説明したため詳細説明は省略する。

また、64KB専用領域管理テーブル40の1つのエントリ41の内の情報は第3A図のエントリ51と全く同じである。

尚、第3B図は64KB領域管理テーブル60の1つのエントリ61内の情報を示す図であるが、64KB仮想ページが割り当てられる場合には第1実施例とはセットするフィールドが異なる。すなわち、フィールド97、98、99は使用しない。

次に、第1実施例に対して追加された各キューについて述べる。

第10A図は未割り当て4KB共用キューを示す。このキューは第1実施例のキュー720（第4B図）と同形式であり、実記憶10Aの4KB

キューポイント173とから形成される。各エントリのキューのされ方は前記キュー910と同様であり省略する。なお、このキューがシステム全体で唯一個のキューを形成するのは、前記キュー910とは異なりこのエントリ61H内はもともと16個の4KB実ページ21A群により構成されるため、キー記憶がエントリ51H対応に存在し、各々のエントリ51Hに対応する4KB実ページは異なる記憶キーの仮想ページやジョブに割り当て可能であるためである。第10C図は未割り当て64KBページ専用キューを示す。このキューは64KBページ専用領域30A上の仮想ページが割り当てられていないテーブル40のエントリ41の列とキューポイント175とから形成される。キューのされ方は第1実施例のキュー710と同様であり省略する。第11A図は割り当て済み64KBページキューを示す。このキューはジョブ別に設けられたキューで、実ページサイズを問わずそのジョブの64KB仮想ページに割り当てられている管理テーブル40のエントリ

41 (領域30A上)の列および管理テーブル60のエントリ61J (領域20A上)、61I (領域30上)の列とキューポインタ176とから形成される。

次に実記憶割り当てプログラム2Aの動作を説明する。

第12A、12B図はページ割り当てプログラム2Aの概略フローを示す図である。あるプログラム(ジョブ)の実行中に仮想記憶上のある領域をアクセスしたとき、その仮想領域に対応する実記憶領域が割り当てられていない場合に、ページフォールト割り込みが発生し、これを契機に仮想領域に実記憶領域を仮想ページ単位に割り当てるための実記憶割り当て要求がおこり、実記憶割り当てプログラム2Aが起動されるのは第1実施例と同様である。

実記憶割り当てプログラム2Aが起動されると、まずステップ2000でページフォールト割り込みが発生した仮想アドレスを調べる。仮想アドレスが2GB以上であれば後で説明する第12B図

するが、この場合も同様に最初はキューが存在しないためステップ2100は失敗するのでこの処理についても後に説明する。次に未割り当て4KB共用キュー1720 (第10A図) から未使用ページをサーチするためステップ2200でそのキューポインタ172がゼロであるか否かを調べゼロでなければ、ステップ2210でこのキュー1720の先頭に接続されるエントリ61Gを選択する。さらに、このエントリ61Gに連なる16個の4KBページ21A群に対するエントリ51Gの列の先頭のエントリを選択しそれに対応する4KBページ21Aを割り当て対象とする。次に、ステップ2220で管理テーブルエントリおよびキューの更新を行う。すなわち、エントリ61Gの内部ポインタF (第3B図) で示されるエントリ61Gに連なる先頭のエントリ51Gを選択し、そのフィールド91A、92A、(第3図A) に割り当て対象となった仮想ページのアドレスと割り当て要求元のジョブのIDをセットし、選択したエントリ61G内の割り当て済み分割領

域に示す64KB仮想ページの割り当てを実施する。仮想アドレスが2GB以下であれば4KB仮想ページの割り当てを実施する。4KB仮想ページの割り当ての場合には、まず、ステップ210で第1実施例で述べた要求元ジョブ割り当て済み64KBページ内サーチ処理を実施する。この処理の詳細は前述の通りであり省略するが、この処理は64KBページ内を分割使用した場合に生じる64KBページ内の部分未割り当て分割領域を利用するための工夫であり、要求元ジョブに既に部分的に割り当て済みの64KBページ内の未割り当て分割領域を優先してそのジョブに割り当てる。ジョブに最初にページを割り当ててる場合はそのジョブに対して部分的に未割り当ての64KBページは存在しない。従って、サーチ210は失敗するのでこのサーチ210については後に説明する。ステップ2010でサーチ210において部分的に未割り当てのページが見つからなかったことを判断すると、次に、部分割り当て済み4KB共用キュー (第10B図) から未使用ページをサーチ

域数96を+1し(その初期値は0である)する。

次に、キューの更新を以下のように行う。即ち、未割り当て4KBページ共用キュー1720から(第10A図) 上記選択されたエントリ61Gおよびこれに連なる16個のエントリ51Bをはずし、このエントリ61Gおよび上記16個のエントリ51Gの中の、上記選択されたエントリ51G以外の15個(未割り当ての4KBページ21A群に対するテーブル50のエントリ)を部分割り当て済み4KBページ共用キュー1730 (第10B図) に接続する。さらに、上記選択されたエントリ51Gをそのジョブに対して設けられた割り当て済み4KBページキュー810 (第5A図) に接続する。

さらに割り当て対象となった仮想ページに指定されている記憶キー(ジョブから明示的に指定されていない場合は、そのジョブのその時点のプログラムキー値)を割り当て対象となったエントリ51Gに対応するキー記憶5 (第1図) のエントリ6のSK7にセットする。なお、この4KB実ベ

ージのアドレスは、選択されたエントリ51Gが管理テーブル50中の何番目のエントリであるか、すなわち、エントリ51Gのテーブルインデックスにより決定することができる。この4KB実ページアドレスに基づいてハードウェア命令を用いて記憶キーセットする。

次に、ステップ2600により、実記憶割り当てが要求された仮想ページに対応するアドレス変換テーブル内のエントリに選択した分割領域の実アドレスのセットとこのエントリのバリデーションを実施する。こうしてこの実記憶領域が使用可能となる。こうして、未割り当て4KBページ共用キュー1720内の最初の割り当てが終了する。

以上の処理における、ステップ2200でキューポインタ(172)がゼロであれば領域20A上には未割り当てページは無しと判断し、次に未割り当て64KBページキュー720(第4B図)から未割り当て64KBページのサーチを実施し、未割り当て64KB内の一つの分割領域を割り当てる。すなわち、未割り当て64KBページキュー

み分割領域数96を+1し(その初期値は0である)、フィールド92BにそのジョブのジョブIDをセットする。

次に、キューの更新を以下のように行う。即ち、未割り当て64KBページキュー720から(第4B図)上記選択されたエントリ61Bおよびこれに連なる16個のエントリ51Bをはずし、このエントリ61Bおよび上記16個のエントリ51Bの中の、上記選択されたエントリ51B以外の15個(未割り当ての分割領域に対するテーブル50のエントリ)をそのジョブに対して設けられた部分割り当て済み64KBページキュー910(第5C図)に接続する。さらに、上記選択されたエントリ51Bをそのジョブに対して設けられた割り当て済み分割領域キュー820(第5B図)に接続する。さらに、割り当て対象となった仮想ページに指定されている記憶キー(ジョブから明示的に指定されていない場合は、そのジョブのその時点のプログラムキー値)をフィールド97にセットし、割り当て対象となった分割領

域720(第4B図)を用いて、そのジョブに割り当てべき未使用の64KBページを見つける。このために、まずステップ2300でこのキュー720のポインタ72の値のゼロ判定を行いこのポインタに接続されたエントリ61Bの有無調べる。

キューポインタ72がゼロでなければ、ステップ2310にてそれに連なるキュー720の先頭のエントリ61Bを選択する。そのエントリ61Bに連なる16個の分割領域に対するエントリ51Bの列の先頭のエントリを選択しそれに対応する分割領域を割り当て対象とする。

次に、ステップ2320で管理テーブルエントリおよびキューの更新を行う。すなわち、エントリ61Bの内部ポインタF(第3B図)で示されるエントリ61Bに連なるエントリ51Bを選択し、前記と同様にそのフィールド91A, 92A,

(第3図A)に割り当て対象となった仮想ページのアドレスと割り当て要求元のジョブのIDをセットし、選択したエントリ61B内の割り当て済

域の含まれる64KB実ページに対応するキー記憶5(第1図)のエントリ6にフィールド97にセットした値と同じキー値をセットする。なお、この64KB実ページのアドレスは選択されたエントリ61Bがテーブル60中の何番目のエントリであるか(すなわち、それに対応して定まることエントリ61Bのテーブルインデックス)から決定することができる。この64KB実ページのアドレスに基づいてハードウェア命令を用いて上記記憶キーをキー記憶5の対応するエントリ6のフィールドSK7(第8図)に書き込む。また、この記憶キーの書き込みは未割り当て64KBページ内から分割領域を最初に割り当てる場合のみ実施する。次に、前記同様にステップ2600により、実記憶割り当てが要求された仮想ページに対応するアドレス変換テーブル内のエントリに選択した分割領域の実アドレスのセットとこのエントリのバリデーションを実施する。こうしてこの実記憶領域が使用可能となる。こうして、未割り当て64KBページキュー910からの最初の分

割領域の割り当てが終了する。

上記の処理における、ステップ2300においてキューポインタ(72)がゼロであればキュー910にはエントリが存在しないと判断し未割り当て4KBページキュー710からの割り当てを試みる。この処理は実記憶10Aの4KB実ページ専用領域20内の未割り当ての一つの実ページを割り当てるべき実記憶領域として選択する。この選択は、未割り当て4KBページキュー710(第4A図)に接続されたエントリが有るかどうかをステップ2400で調べ、有ればステップ2410でキュー710の先頭につながるエントリ51Aを選択することにより行われる。次にステップ2420において、この選択されたエントリ51A内に割り当て要求もとのジョブのIDおよび割り当て対象の仮想ページのアドレスを書き込む。さらに、このエントリ51Aを上記キュー710よりはずし、そのジョブに対して設けられた割り当て済み4KBページキュー810(第5A図)に接続する。さらに、前記エントリ51G

にそのジョブのIDおよび割り当て対象となった仮想ページのアドレスをセットする。

さらに、上記エントリ61E内の割り当て済み分割領域数を+1する。

上記エントリ51Eを上記キュー910からはずし、そのジョブに対して設けられている割り当て済み分割領域キュー820に接続する。もし、上記更新後の割り当て済み分割領域数が16に等しいときは、割り当て対象となった分割領域を含む64KB実ページは未割り当ての分割領域がなくなったことになる。したがって、この時には、この64KB実ページに対応する前記エントリ61Eもキュー910からはずし、全割り当て済みキュー920(第5D図)にエントリ61Fとして接続する。こうして、ステップ234Cがするとその後ステップ2600が前と同様になされる。上記のステップ2010で割り当て可能な部分未割り当ての64KBページが無かったか場合には、次に前記の説明で後回しにしたもう一つの処理、すなわち、部分割り当て済み4KBページ

の場合と同様に記憶キーのセットを行い、ステップ2600においてアドレス変換テーブルの書き換えを先に述べたのと同じように行う。こうして4KB実ページへの割り当てが終了する。その後、4KB仮想ページに対する新たな実記憶割り当て要求がなされる毎に同じように処理される。この際、先に詳細を省略した要求元ジョブ割り当て済み64KBページサーチ210も実行される。以下、この処理について述べる。

ステップ210の処理は既に第1実施例で詳述したので省略するが、ステップ210が完了すると、ステップ2010でこのジョブに設けられた部分割り当て済み64KBページキュー910内に割り当て可能な部分未割り当ての64KBページが有ったか無かったかを判断し、有ればステップ234Cにてそのエントリおよびキューの更新を行う。すなわち、そのエントリ61Eに接続されてた未使用分割領域のエントリ51Eを一つ選択し、このエントリに対応する分割領域を割り当て領域とすべく、この選択されたエントリ51E

共用キュー1730(第10B図)からの割り当てを行う。この処理はステップ2100でこのキューのポインタ173がゼロでない場合に実施され、ステップ2110でキュー1730の先頭のエントリ61Hを取り出す。そのエントリ61Hに連なる4KBページ21A群に対するエントリ51Hの列の先頭のエントリを選択しこれを割り当て対象とする。

次に、ステップ2120で管理テーブルエントリおよびキューの更新を行う。すなわち、エントリ61Hの内部ポインタF(第3B図)で示されるエントリ61Hに連なるエントリ51Hを選択し、そのフィールド91A、92A、(第3図A)に割り当て対象となつた仮想ページのアドレスと割り当て要求元のジョブのIDをセットし、選択したエントリ61B内の割り当て済み分割領域数96を+1し(その初期値は0である)する。さらに前記エントリ51Gの場合と同様に記憶キーをセットする。次に、上記エントリ51Hを上記キュー1730よりはずし、そのジョブに対して

設けられた部分割り当て済み4KBページキュー810(第5A図)に接続する。もし、上記更新後の割り当て済み分割領域数が16に等しい時は、上記エントリ61H内の全ての領域が使用されたことを表す。したがって、この場合には前記エントリ61Hもキュー1730からはずす。こうして、ステップ2120が終了するとその後のステップ2600が前と同様になされる。

また、上記のいずれの処理においても割り当て可能な領域が無い場合には後で述べるページアウトプログラム7A(第14A, 第14B図)が動作し、実記憶に使用可能なページを一定量確保できるまでステップ2500で当該割り当て要求元ジョブを停止させて実記憶の空きを待たせる。この場合は、ページアウトプログラム7Aに対していずれのサイズの仮想ページを割り当てるための領域が不足しているかを通知する。以上の処理を実ページ割り当て要求がなされるたびに繰り返せばよい。

次に、前記ステップ2000でページフォールト



テップ3500により、実記憶割り当てが要求された仮想ページに対応するアドレス変換テーブル内のエントリに選択したエントリ41に対する実アドレスのセットとこのアドレス変換テーブルエントリのバリデーションを実施する。

こうしてこの実記憶領域が使用可能とする。こうして、未割り当て64KBページ専用キュー1750内のからの割り当てが終了する。

以上の処理における、ステップ3000でキューポインタ(175)がゼロであれば領域30A上には未割り当てページは無しと判断し、次に未割り当て64KBページキュー720(第4B図)から未割り当て64KBページのサーチを実施し、未割り当て64KBの一つを割り当てる。すなわち、未割り当て64KBページキュー720(第4B図)を用いて、そのジョブに割り当てるべき未使用の64KBページを見つける。このために、まずステップ3100でこのキュー720のポインタ72の値のゼロ判定を行いこのポインタに接続されたエントリ61Bの有無調べる。

発生アドレスが2GB以上の場合、即ち、64KB仮想ページの割り当て処理について第12B図を用いて説明する。64KB仮想ページの割り当て処理では、まず、未割り当て64KBページ専用キュー1750(第10C図)から未使用ページをサーチするためステップ3000でそのキューポインタ175がゼロであるか否かを調べゼロでなければ、ステップ3010でこのキュー1750の先頭に接続されるエントリ41選択し、これを割り当て対象とする。次に、ステップ3020でテーブルエントリおよびキューの更新を行う。すなわち、そのエントリ41フィールド91A, 92A, (第3図A)に割り当て対象となった仮想ページのアドレスと割り当て要求元のジョブのIDをセットする。次に、キューの更新を以下のように行う。即ち、未割り当て64KBページ専用キュー1750から(第10C図)上記選択されたエントリ41をはずし、そのジョブに対して設けられた割り当て済み64KB(仮想)ページキュー176(第11図)に接続する。次に、ス



キューポインタ72がゼロでなければ、ステップ3110にてそれに連なるキュー720の先頭のエントリ61Bを選択し、それに対応する領域を割り当て対象とする。次に、ステップ3120で管理テーブルエントリおよびキューの更新を行う。すなわち、エントリ61Bのフィールド91B, 92B, (第3図B)に割り当て対象となった仮想ページのアドレスと割り当て要求元のジョブのIDをセットし、選択したエントリ61B内の割り当て済み分割領域数96を+16(その初期値は0である)する。

次に、キューの更新を以下のように行う。即ち、未割り当て64KBページキュー720から(第4B図)上記選択されたエントリ61Bをはずし、そのジョブに対して設けられた割り当て済み64KBページキュー1760(第11図)に接続する。次ぎに、前記同様にステップ3500により、実記憶割り当てが要求された仮想ページに対応するアドレス変換テーブル内のエントリに選択した領域の実アドレスをセットしアドレス変換テー

ルエントリのバリデーションを実施する。こうしてこの実記憶領域が使用可能となる。

上記の処理における、ステップ3100においてキューポインタ(72)がゼロであればキュー910にはエントリが存在しないと判断し未割り当て4KBページ共用キュー1720(第10A図)からの割り当てを試みる。この処理は実記憶10Aの4KB実ページ共用領域20A内の連続する16個の4KBページ21A群よりなる未割り当ての64KB領域を割り当てるべき実記憶領域として選択する。この選択は、未割り当て4KBページ共用キュー1720に接続されたエントリが有るかどうかをステップ3200で調べ、有ればステップ3210でキュー1720の先頭につながるエントリ61Gを取り出すことにより行われる。次に、ステップ3220で管理テーブルエントリおよびキューの更新を行う。すなわち、エントリ61Gのフィールド91B, 92B, (第3図B)に割り当て対象となった仮想ページのアドレスと割り当て要求元のジョブのIDをセ

ットし、選択したエントリ61B内の割り当て済み分割領域数96を+16(その初期値は0である)する。

次に、キューの更新を前記同様に行う。即ち、未割り当て4KBページ共用キュー1720から(第10A図)上記選択されたエントリ61Gをはずし、そのジョブに対して設けられた割り当て済み64KBページキュー1760(第11図)に接続する。次に、前記同様にステップ2350においてアドレス変換テーブルの書き換えを先に述べたのと同じように行う。こうして64KB仮想ページに対する実記憶領域の割り当てが終了する。その後、64KB仮想ページに対する新たな実記憶割り当て要求がなされる毎に同じように処理される。

また、上記のいずれの処理においても割り当て可能な領域が無い場合には前記同様に未使用ページを一定確保できるまでステップ2500で当該割り当て要求元ジョブを停止させて実記憶の空きを待たせる。

この場合は、ページアウトプログラム7Aに対して64KBのサイズの仮想ページを割り当てるための領域が不足していることを通知する。以上の処理を64KB仮想ページに対する実ページ割り当て要求がなされるたびに繰り返せばよい。

こうして、仮想ページサイズと実ページサイズが同一の場合は元よりサイズの異なった仮想領域と実記憶領域間の相互割り当てができる。

次にページ計測プログラム3(第9図)の動作を説明する。第7A, 7B図はページ計測プログラム3の概略フローを示す図である。第1実施例では点線で示したステップ370, 380は使用しなかったが本実施例では、これらを含め全てを使用する。以下、ページ計測プログラム3の第2の実施例における動作を説明する。

本実施例においてもページ計測処理は、仮想ページとそれに割り当てられた実記憶が属する実ページサイズが異なる場合、その仮想ページと実ページのサイズの大きいほうの領域毎にページ計測を行う所にあることは第1実施例と同じである。

以下、詳細説明を行う。ステップ310~340, およびステップ390, 400は既に第1実施例で詳細を述べたのでこれらについては簡単化のため詳しくは述べない。まず、ページ計測プログラム3は、ページ計測すべきジョブに対して設けられた全割り当て済み64KBページキュー920(第5B図)の先頭ポインタ92を変数QUEUEにセットし、ステップ320で内部サブルーチンSUB1をコールする。

このキュー920は4KBの仮想ページが割り当てられている64KB実ページ領域30に対するテーブル60のエントリを表すキューの一つで前述の64KB実ページ内の全分割領域が各々16個の仮想ページに割り当て済みである64KBページを示し、上記の処理ではこれらのページに対するページ計測を64KB単位に行ったことになる。

次に、64KBページ領域30上に割り当てられている4KB仮想ページの内、もう一つの部分に割り当て済み64KBページキュー910(第5

C図)に接続されているものについて、ステップ330、340で同様に64KB単位にページ計測を実施する。次に、本実施例で初めて現れる64KB仮想ページに割り当てられている領域のページ計測を実施する。64KB仮想ページに割り当てられている実記憶領域は第12B図の64KB写ページの割り当てで述べた様に実記憶上の領域の種類によらず全てが第11図に示すキュー1760に接続されている。したがって、第7A図に示すように、このキュー1760のキューポインタ176をステップ370において変数QUEUEにセットし、ステップ380で内部サブルーチンSUB1をコールすれば全ての64KB仮想ページに割り当てられている領域のページ計測が行える。すなわち、サブルーチンSUB1はコールされると、ステップ510において、変数QUEUEの値がゼロか否かによりページ計測すべきエン트리41、エン트리61I、エン트리61Jのいずれかがあるか否かを調べる。次にエン트리がある場合にはステップ440により変数

QUEUEで指されるエン트리41A、エン트리61I、エン트리61Jのいずれかに対応する64KB実領域の使用の有無を調べる。これは、第1の実施例と同様にハードウェア命令により第1図に示すキー記憶エン트리6上の参照ビットR7を読み出すことにより行う。

これらのビットがゼロであれば当該ページは、前回のページ計測時点から今回のページ計測までの間使用されなかったことが分かる。次に、当該ページが使用されていれば、ステップ450において不参照期間数93A(第3A図または第3B図)をクリアし、ステップ460で当該エン트리(エン트리41A、エン트리61I、エン트리61Jのいずれか)を当該キュー1760の最終位置に移動する。このエントリの移動は、不参照期間数93Aの大きい順にキューを構成するためである。次に、当該ページが使用されていなければステップ470において不参照期間数をカウンタアップする。



このように、ページの使用頻度はページ計測イ

ンターバル毎にそのページが使用されていない場合に不参照期間数をカウントアップし、使用されていた場合には不参照期間数をゼロにするため、この不参照期間数が大きいほど使用頻度が低いことを示す。

次にステップ480により変数QUEUEに現在処理しているエン 트리内のフォワードポインタ(FWP2)QUEUEにセットし、新たなエン 트리に対して今まで述べてきたステップ510~480の処理を変数QUEUEの値がゼロすなわち、キュー1760に接続された全てのエントリを処理し終わるまで繰り返す。以上で、64KB仮想ページに割り当てられている全ての実施例に対して64KB単位にページ計測を実施した。最後に、4KBページ領域20、および20A上に割り当てられている4KB仮想ページに対してページ計測を実施する。この場合には、ページ計測処理は4KB単位に実施する。4KBページ領域20、および20A上に割り当てられている4KB仮想ページは全て、割り当て済み4KBページ

キュー810(第5A図)に接続されている。この場合も、前記同様にステップ390で変数QUEUEにこのキューの先頭エン 트리51Cを示すポインタ81をセットしステップ400で内部サブルーチンSUB1をコールする。サブルーチンSUB1は、前記キュー810に接続されているエン 트리51Cを対象として前記同様にステップ510~480の処理を繰り返すことにより当該キュー上に接続されるエン 트리51Cに対するページ計測処理を4KB単位に行う。以上の処理を実行中のジョブがCPUがある一定時間使用するつど実施すればよい。こうして、第2実施例においてもページ計測処理はその仮想ページサイズとその仮想ページに割り当てられている実記憶の実ページサイズの大きいほうのサイズでできページ計測数を低減することができる。

次にページアウトプログラム7A(第9図)の動作を説明する。

第14図はページアウトプログラム7A概略フローを示す図である。本実施例におけるページア

ウト処理も第1実施例と同様に、仮想ページとそれに割り当てられた実記憶が属する実ページサイズが異なる場合、その仮想ページと実ページサイズの大きいほうのサイズ単位にページアウトを行う所にある。以下、図にしたがって動作を説明する。

第1実施例と同様に、本ページアウトプログラム7Aは実記憶の使用可能な領域が、ある一定量以下になると起動され、また、不参照期間数がnに以上のページをNPAGE個ページアウトするべく起動されるものとする。ページアウトプログラム7Aが起動されると、まず、ステップ400でページアウトしたページ数をカウントする変数POUTを0にする。次に、ステップ4100でいずれのサイズの仮想ページ割り当て用の領域を処理するのかを判断する。これは、ページアウトプログラム7Aが起動されるときに起動者により指定される。仮想4KB用の処理であれば、第1実施例で述べたページアウトプログラム7(第13A図、13B図)を実行し処理する。この詳細

ため、ステップ4760において変数RPNに当該エントリ(エントリ41A、エントリ61I、エントリ61Jのいずれか)のフォワードポインタをセットとステップ4400からの処理を繰り返す。上記のステップ4410でページアウト可能であれば、ステップ4500にて、当該エントリの不参照期間数がページアウトプログラム起動時に指示されたn以上かを調べる。n以下であれば、ページ計測プログラム3Aで述べたように各エントリは不参照期間数が大きい順に並んでいるため以降のエントリ61Eもn以下であり、当該キューには今回のページアウト起動条件を満たすページが無いためリターンする。

n以下であれば、次に、ステップ4600で当該エントリに対応する64KBの実記憶領域をページアウトし(但し当該エントリに対応するキー記憶上のチェンジビットC(第8図5)がoffの場合はページアウトは行わない)仮想ページに対応するアドレス変換テーブルのエントリを無効とする。次にステップ4700にて当該エントリ

は説明済みである。ステップ4100で仮想64KB用の処理であると判断すると、次にステップ4300でページアウトすべきジョブに対して設けられた割り当て済み64KBページキュー1760(第11図)の先頭ポインタ176を変数RPNにセットし、ページアウトすべきキュー1760の先頭エントリ(エントリ41A、エントリ61I、エントリ61Jのいずれか)をセットする。その後ステップ4400において、変数RPNの値がゼロか否かによりページアウトすべきエントリあるか否かを調べ、エントリが無ければ、何もせずリターンする。次にエントリがある場合には、次にステップ4410でページアウトして良いページかどうかを調べる。これは、そのページのページアウトが禁止されている(ページ固定されている)とか、既に前回のページアウト処理でページアウト中であると言った従来からのチェックで、仮想記憶方式の計算機において通常成されているチェックである。ページアウト不可であれば、キューに接続された次のページを処理する

(エントリ41A、エントリ61I、エントリ61Jのいずれか)がいずれの実記憶領域に属するかを調べる。これは、第8図述べた実記憶10Aの4つの領域(領域20、領域20A、領域30、領域30A)の各々の大きさがシステムの初期化(IPL)時に定まる様に成っており、各領域の先頭および末尾の実アドレスがあらかじめ分かっている。このアドレスと当該エントリのポインタである変数RPNとを比較することにより当該エントリが属する領域が分かる。当該エントリが領域30Aに属している場合には、ステップ4710が実行されキュー1760(第11図)よりエントリ41Aをはずし、未割り当て64KBページ専用キュー1750(第10C図)へ接続する。次にステップ4740でページアウト数を示す変数POUTを1増し、ステップ4750でPOUTがページアウトすべき個数に達したか否かをチェックするため変数NPAGEと比較する。ページアウト数がNPAGE未満であれば、つぎにステップ4746で変数RPNに当該エン

トリ41Aのフォワードポインタをセットしステップ4400からの処理を繰り返す。

また、前記ステップ4700で領域30に属する場合にはステップ4720が実行されキュー1760よりエントリ61Iをはずし、さらに、当該エントリ61I上の割り当て済み分割領域数96(第3B図)を0にセットし、さらに、エントリ61Iを未割り当て64KBページキュー720(第4B図)に接続する。次に、前記同様にステップ4740でページアウト数を示す変数POUTを1増し、ステップ4750でPOUTがページアウトすべき個数に達したか否かをチェックするため変数NPAGEと比較し、ページアウト数がNPAGE未満であれば、つぎのステップ4746で変数RPNに当該エントリ61Iのフォワードポインタをセットしステップ4400からの処理を繰り返す。

また、前記ステップ4700で領域20Aに属する場合にはステップ4730が実行されキュー1760よりエントリ61Jをはずし、さらに、

当該エントリ61J上の割り当て済み分割領域数96(第3B図)を0にセットし、さらに、エントリ61Jを未割り当て4KBページ共用キュー1720(第10A図)に接続する。次に、前記同様にステップ4740でページアウト数を示す変数POUTを1増し、ステップ4750でPOUTがページアウトすべき個数に達したか否かをチェックするため変数NPAGEと比較し、ページアウト数がNPAGE未満であれば、つぎにステップ4746で変数RPNに当該エントリ61Jのフォワードポインタをセットしステップ4400からの処理を繰り返す。以上に処理をステップ4400で当該キューの全エントリを処理し終わるか、ステップ4500で起動時の条件以下の不導照期間数を持つエントリが現れるまで、もしくは、ステップ4750で所定のページ数(NPAGE)をページアウトするまで繰り返す。

以上で、64KB仮想ページに割り当てられている実記憶領域を64KB単位にページアウトを実施し処理を終了する。以上の処理で十分な認使

用実記憶領域が得られなかった場合には、ページアウトプログラム7A起動時の1つの条件であるページアウトすべき不参照期間数を順次小さくして繰り返しページアウトプログラム7Aを起動すれば良い。こうして、仮想ページサイズ、および実ページサイズが各々2種存在する本第2実施例におけるページアウト処理を仮想ページサイズとその仮想ページに割り当てられた実記憶の実ページサイズの内大きいほうのサイズで実施することができる。

本実施例によれば仮想ページサイズと実ページサイズが必ずしも一致しなくてもよく、自由度の高い実記憶り拡張が可能となり、実ページサイズを64KBに拡張することにより従来の4KBに比べてオペレーティングシステムで管理する実ページ数が1/16となり管理オーバーヘッドの削減ができる。

〔発明の効果〕

本発明によれば、実記憶/仮想記憶上の異なるサイズにより構成された各領域間で相互にページ

の割り当てが可能となるため、主記憶の効率的な管理が可能となる。また、ページ計測処理を仮想ページサイズ/実ページサイズの内大きい方の単位で実施するため計測数が減りオーバーヘッドの大幅削減が図れる。

4. 図面の簡単な説明

第1図は本発明の第1の実施例の仮想記憶と実記憶の構成図、第2図は本発明の第1実施例における構成図と実記憶の対応図、第3A、3B、~~4A~~4A、4B、5A~5D図は本発明の第1実施例における各種実記憶管理テーブルの説明図、第6図は第1実施例におけるページ割り当てプログラムの処理フロー図、第7A、7B図は本発明の第1実施例、第2実施例におけるページ計測プログラムの処理フロー図、第8図は本発明の第2の実施例の仮想記憶と実記憶の構成図、第9図は本発明の第2実施例における構成図と実記憶の対応図、第10A~10C、11図は第2実施例における、各種実記憶管理テーブルの説明図、第12A、12B図は第2実施例におけるページ割

り当てプログラムの処理フロー図、第13A、
13B図は本発明の第1実施例におけるページア
ウトプログラムの処理フロー図、第14図は第2
実施例におけるページアウトプログラムの処理フ
ロー図。

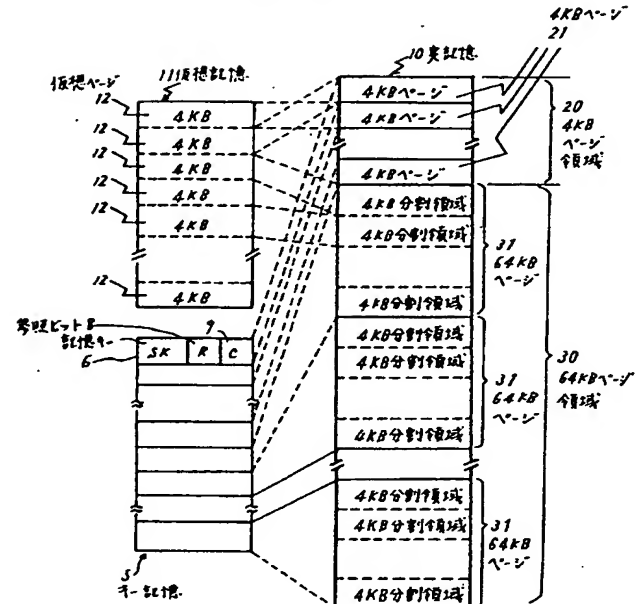
符号の説明

2…ページ割り当てプログラム、3、3A…ペ
ージ計測プログラム、10、10A…実記憶、
11…仮想記憶、20…4KB実ページ領域、
30…64KB実ページ領域、50…第1管理テ
ーブル(4KB単位)、60…第2管理テーブル
(64KB単位)、21…4KB実ページ、
31…64KB実ページ、51…4KB単位の第
1管理テーブルのエントリ、61…64KB単位
の第2管理テーブルのエントリ、7、7A…ペ
ージアウトプログラム。

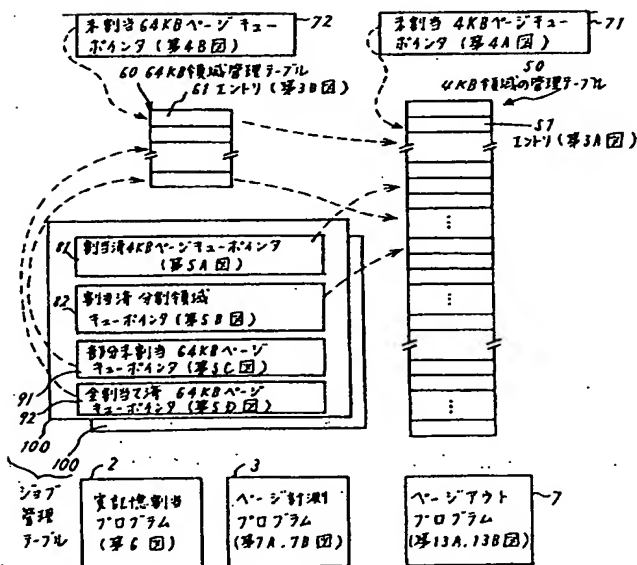
代理人 井理士 小川勝男



第1図

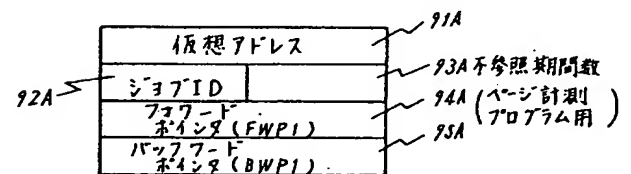


第2図



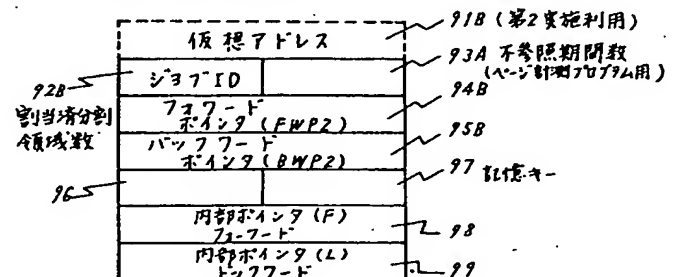
第3A図

51 テーブル50のエントリ



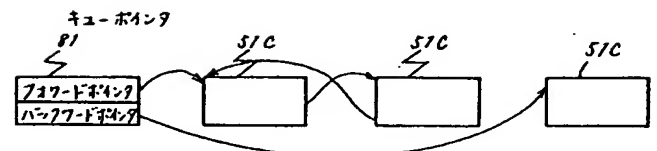
第3B図

61 テーブル60のエントリ



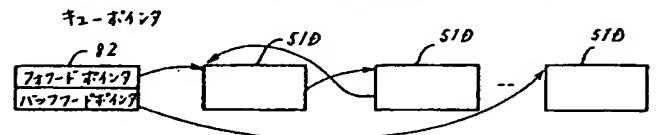
第 5 A 図

810 割当済4KBページキュー(ジョブ別)



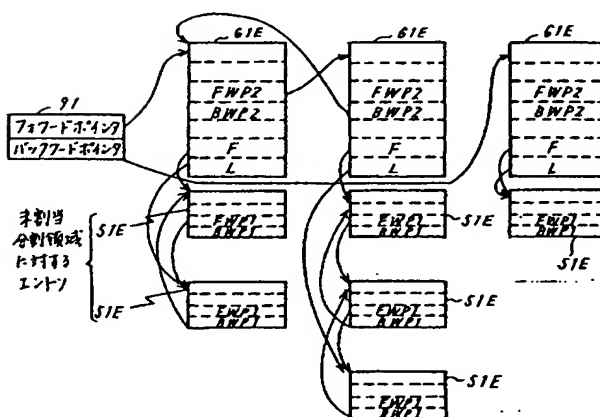
第58回

820 割当済分割領域キュー (ジョブ別)



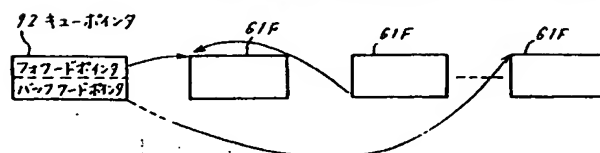
第 5 C 圖

910 部分割当済KBページキュー(ジョブ別)



第 5 D 図

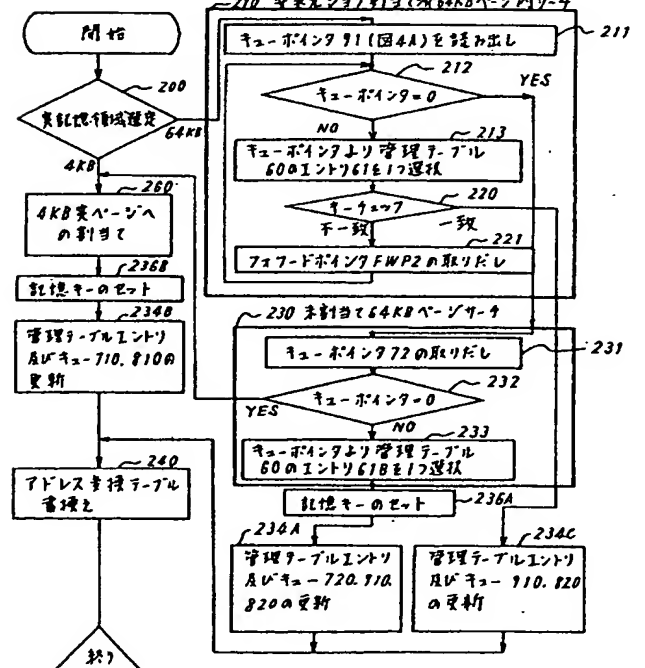
920 全割り当て済64KBページキュー(ジョブ別)



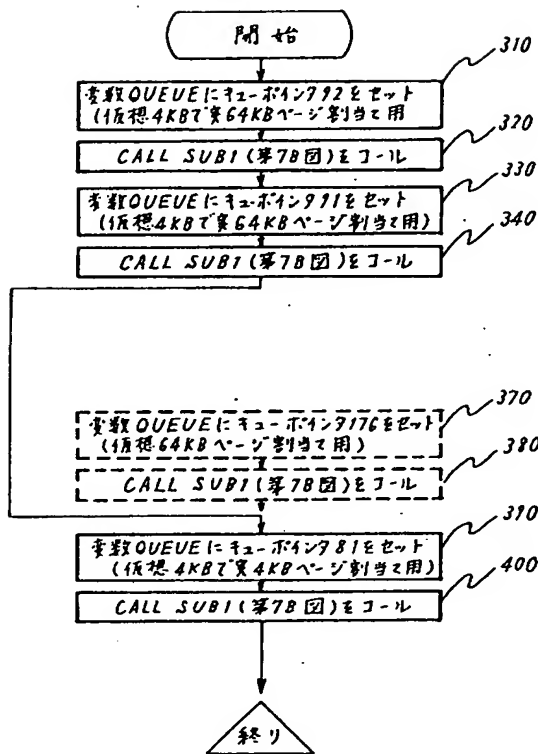
第6圖

2 実証実験とプログラムサ-4

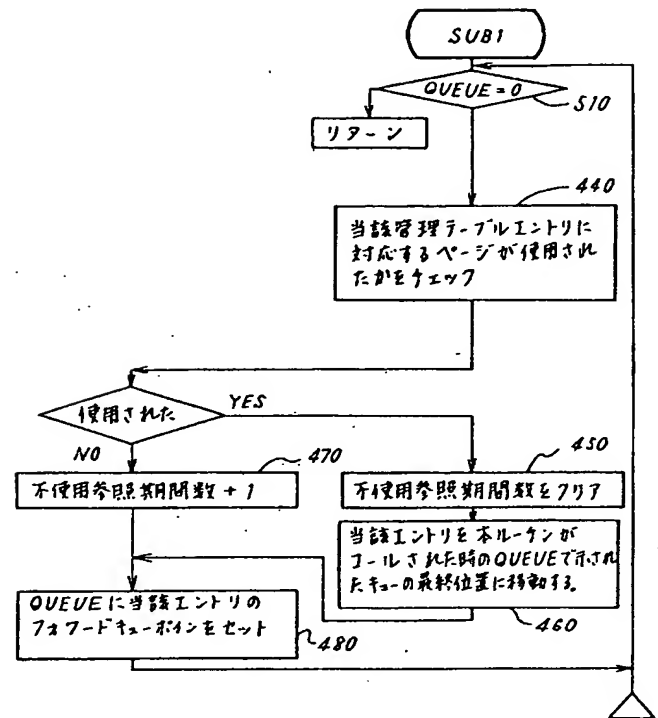
210 要素元ジョブ別で済64KB个-



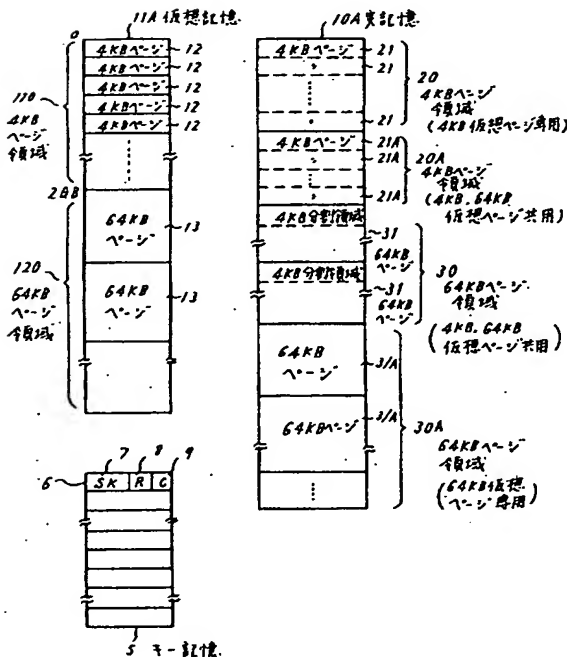
第7A図



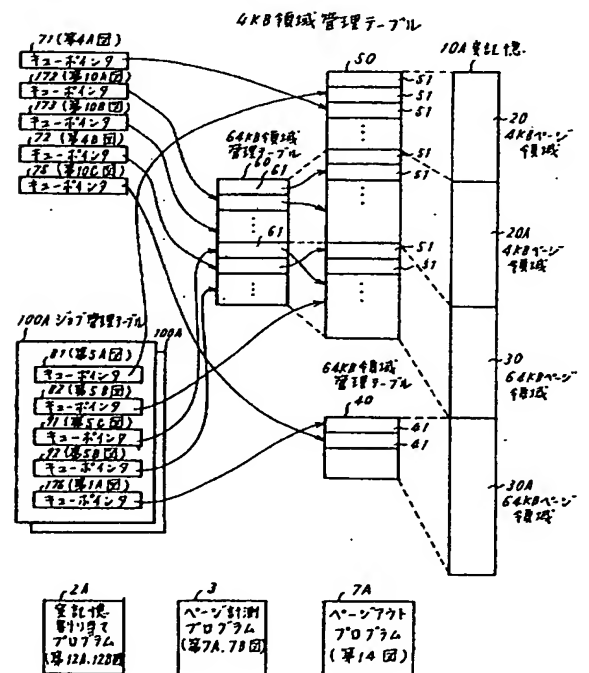
第7B図



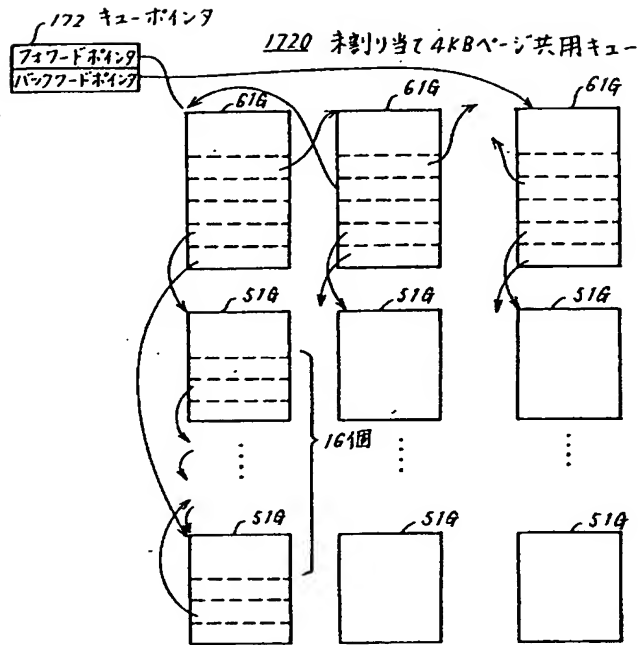
第8図



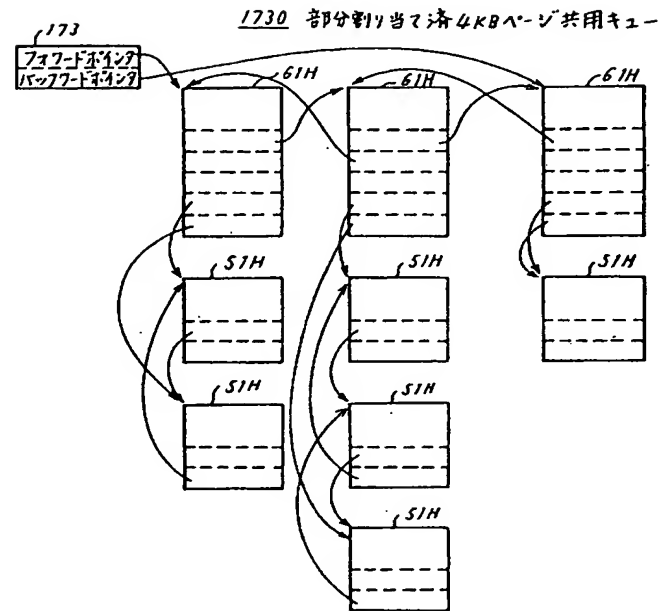
第9図



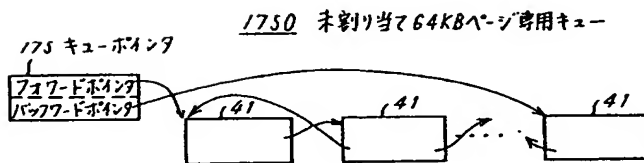
第10A図



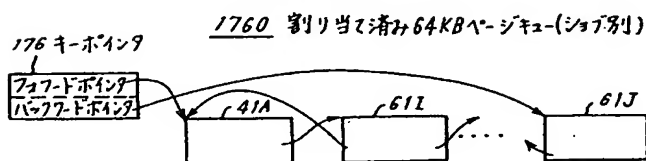
第10B図



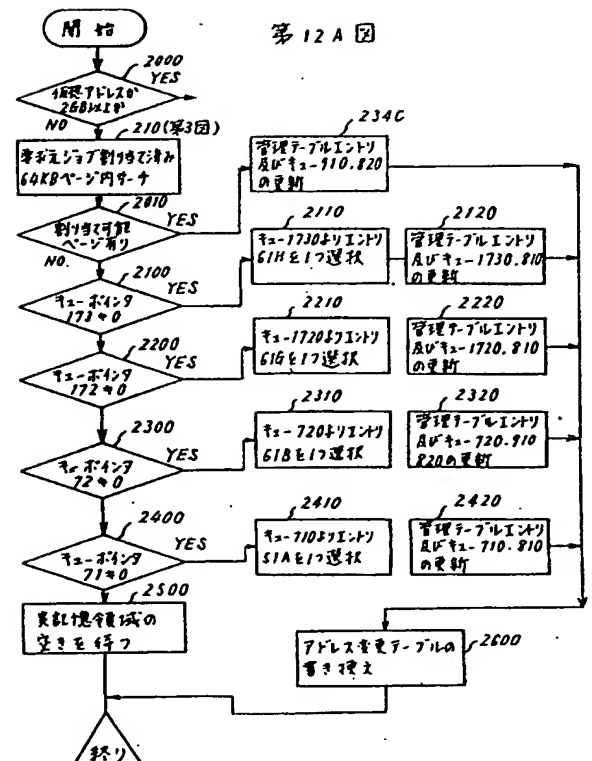
第10C図



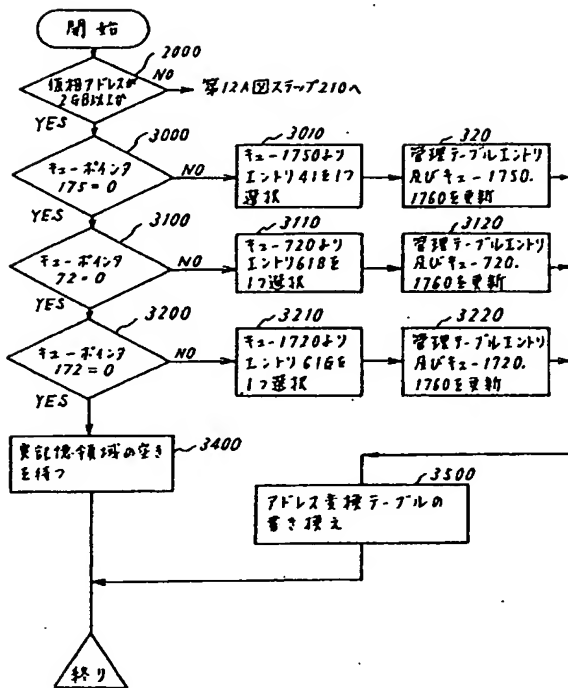
第11図



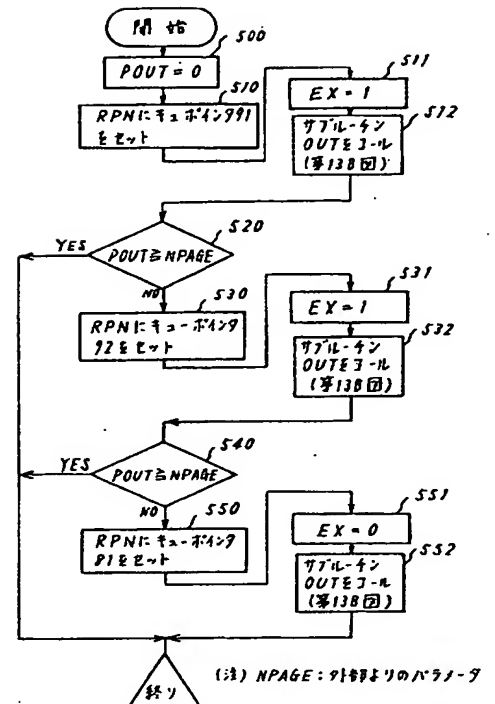
第12A図



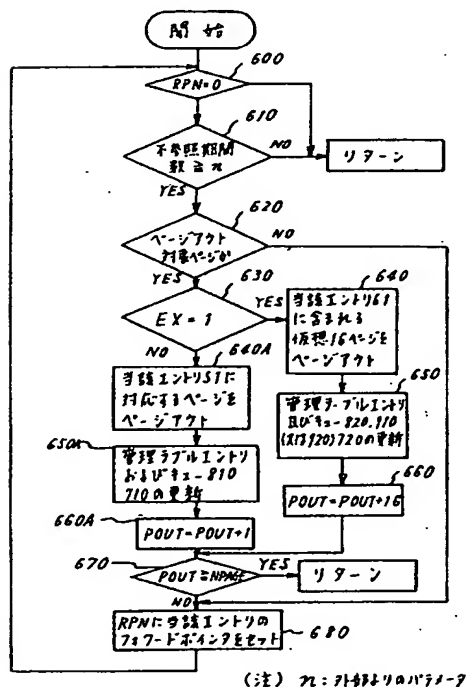
第 12 B 圖



第13人 団



第 13 B 圖



第14圖

